Edited by
Serhii Yevseiev, Yuliia Khokhlachova,
Serhii Ostapov, Oleksandr Laptiev

MODELS OF SOCIO-CYBER-PHYSICAL SYSTEMS SECURITY

Monograph

Technology Center

2023

UDC 004.056
M78

Published in 2023
by PC TECHNOLOGY CENTER
Shatylova dacha str., 4, Kharkiv, Ukraine, 61165

Approved by the Academic Council of National Technical University «Kharkiv Polytechnic Institute», Protocol No. 5 of 23.03.2023

Reviewers:
Dudykevych Valerii, Doctor of Technical Science, Professor, Head of the Department of Information Security of Lviv Polytechnic National University;
Korchenko Alexandr, Doctor of Technical Sciences, Professor, Head of the Department of Information Technology Security of National Aviation University.

M78 **Authors:**
Edited by **Serhii Yevseiev, Yuliia Khokhlachova, Serhii Ostapov, Oleksandr Laptiev**
Serhii Yevseiev, Yuliia Khokhlachova, Serhii Ostapov, Oleksandr Laptiev, Olha Korol, Stanislav Milevskyi, Oleksandr Milov, Serhii Pohasii, Yevgen Melenti, Hrebeniuk Vitalii, Alla Havrylova, Serhii Herasymov, Roman Korolev, Oleg Barabash, Valentyn Sobchuk, Roman Kyrychok, German Shuklin, Volodymyr Akhramovych, Vitalii Savchenko, Sergii Golovashych, Oleksandr Lezik, Ivan Opirskyy, Oleksandr Voitko, Kseniia Yerhidzei, Serhii Mykus, Yurii Pribyliev, Oleksandr Prokopenko, Andrii Vlasov, Nataliia Dzheniuk, Maksym Tolkachov
Models of socio-cyber-physical systems security: monograph / S. Yevseiev, Yu. Khokhlachova, S. Ostapov, O. Laptiev and others. – Kharkiv: PC TECHNOLOGY CENTER, 2023. – 168 p.

The monograph discusses the methodology for cooperative conflict interaction modeling of security system agents. The concept of modeling the structure and functioning of the security system of critical infrastructure facilities is demonstrated. The method for assessing forecast of social impact in regional communities is presented. Counteracting the strategic manipulation of public opinion in decision-making by actors of social networking services based on the conceptual model for managed self-organization in social networking services are developed. Algorithms for thinning the critical infrastructure identification system and their software are implemented.
The monograph is intended for teachers, researchers and engineering staff in the field of cybersecurity, information technology, social engineering, communication systems, computer technology, automated control systems and economic information security, as well as for adjuncts, graduate students and senior students of relevant specialties.
Figures 60, Tables 32, References 132 items.

DOI: 10.15587/978-617-7319-72-5
ISBN 978-617-7319-72-5 (on-line)
ISBN 978-617-7319-73-2 (print)

9 786177 319725

AUTHORS

SERHII YEVSEIEV
Doctor of Technical Science, Professor, Head of Department
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-1647-6444

YULIIA KHOKHLACHOVA
PhD, Associate Professor
Department of Security of Information Technologies
National Aviation University
ORCID ID: https://orcid.org/0000-0002-1883-8704

SERHII OSTAPOV
Doctor of Physical and Mathematical Sciences, Professor, Head of Department
Department of Computer Systems Software
Y. Fedkovych Chernivtsi National University
ORCID ID: https://orcid.org/0000-0002-4139-4152

OLEKSANDR LAPTIEV
Doctor of Technical Science, Senior Researcher
Department of Cyber Security and Information Protection
Taras Shevchenko National University of Kyiv
ORCID ID: https://orcid.org/0000-0002-4194-402X

OLHA KOROL
PhD, Associate Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-8733-9984

STANISLAV MILEVSKYI
PhD, Associate Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0001-5087-7036

OLEKSANDR MILOV
Doctor of Technical Science, Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0001-6135-2120

SERHII POHASII
PhD, Associate Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-4540-3693

YEVGEN MELENTI
PhD, Associate Professor
Special Department No. 5
National Academy of Security Service of Ukraine
ORCID ID: https://orcid.org/0000-0003-2955-2469

VITALII HREBENIUK
Doctor of Science in Law, Senior Reseacher
First Vice-Rector
National Academy of Security Service of Ukraine
ORCID ID: https://orcid.org/0000-0002-5169-8694

ALLA HAVRYLOVA
Senior Lecturer
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-2015-8927

SERHII HERASYMOV
Doctor of Technical Sciences, Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-1810-0387

ROMAN KOROLEV
PhD
Department of Cyber Security and Information Technology
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-7948-5914

OLEG BARABASH
Doctor of Technical Sciences, Professor
Department of Automation of Designing of Energy Processes and Systems
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-1715-0761

VALENTYN SOBCHUK
Doctor of Physical and Mathematical Sciences, Professor
Department of Integral and Differential Equations
Taras Shevchenko National University of Kyiv
ORCID ID: https://orcid.org/0000-0002-4002-8206

ROMAN KYRYCHOK
PhD
Department of Information and Cyber Security named after Professor Volodymyr Buriachok
Borys Grinchenko Kyiv University
ORCID ID: https://orcid.org/0000-0002-9919-9691

GERMAN SHUKLIN
PhD, Associate Professor, Head of Department
Department of Information and Cybersecurity Systems
State University of Telecommunications
ORCID ID: https://orcid.org/0000-0003-2507-384X

VOLODYMYR AKHRAMOVYCH
Doctor of Technical Sciences, Senior Researcher, Professor
Department of Information and Cybersecurity Systems
State University of Telecommunications
ORCID ID: https://orcid.org/0000-0002-6174-5300

VITALII SAVCHENKO
Doctor of Technical Sciences, Professor, Director of Institute
Educational and Scientific Institute of Information Protection
State University of Telecommunications
ORCID ID: https://orcid.org/0000-0002-3014-131X

SERGII GOLOVASHYCH
PhD, Associate Professor
Department of Software Engineering and Management Intelligent Technologies
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0009-0004-2468-1952

OLEKSANDR LEZIK
PhD, Associate Professor
Department of Air Defense Forces Tactics of the Land Forces
Ivan Kozhedub Kharkiv National Air Force University
ORCID ID: https://orcid.org/0000-0002-7186-6683

IVAN OPIRSKYY
Doctor of Technical Science, Professor
Department of Information Security
Lviv Polytechnic National University
ORCID ID: https://orcid.org/0000-0002-8461-8996

OLEKSANDR VOITKO
PhD, Head of Center
Educational and Scientific Center of Strategic Communications in the Field of Ensuring National Security and Defense
Institute of Troops (Forces) and Information Technologies
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0002-4610-4476

KSENIIA YERHIDZEI
PhD
Educational and Scientific Center of Strategic Communications in the field of Ensuring National Security and Defense
Institute of Troops (Forces) and Information Technologies
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0003-4634-133X

SERHII MYKUS
Doctor of Technical Science, Professor
Institute of Information and Communication Technologies and Cyber Defense
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0002-7103-4166

YURII PRIBYLIEV
Doctor of Technical Science, Professor
Department of Information Technologies Employment and Information Security
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0003-1941-3561

OLEKSANDR PROKOPENKO
PhD
Laboratory Detection and Forecasting of Information Threats
Educational and Scientific Center of Strategic Communications in the Field of Ensuring National Security and Defense
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0002-5482-0317

ANDRII VLASOV
PhD, Associate Professor
Department of Information Technology Security
Kharkiv National University of Radioelectronics
ORCID ID: https://orcid.org/0000-0001-6080-237X

NATALIIA DZHENIUK
Associate Professor
Department of Information Systems
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-0758-7935

MAKSYM TOLKACHOV
Associate Professor
Department of Information Systems
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0001-7853-5855

ABSTRACT

The development of technologies and computing resources has not only expanded the range of digital services in all spheres of human activity, but also determined the range of targeted cyber attacks. Targeted attacks are aimed at destroying not only the business structure, but also its individual components that determine critical business processes. The continuity of such business processes is a critical component of any company, organization or enterprise of any form of ownership, which has a critical impact on making a profit or organizing production processes. The proposed concept of determining the security level of critical business processes is based on the need to use multiloop information security systems. This makes it possible to ensure the continuity of critical business processes through a timely objective assessment of the level of security and the timely formation of preventive measures. This approach is based on the proposed rules for determining the reach of a given security level, based on assessments of the integrity, availability and confidentiality of information arrays, as well as computer equipment for different points of the organization's business processes. The issues of applying situational management methods to ensure the safe functioning of objects of socio-cyber-physical systems, logical and transformational rules that form the foundation for building a situational type cybersecurity management system are considered. One of the main tasks of systems of this type is described – the task of replenishing the description of the situation. The use of pseudophysical logics, various types of pseudophysical logics, the method of their construction and their interconnection are proposed. Particular attention is paid to causal pseudophysical logic, as the least developed for the purposes of ensuring cybersecurity. The formation of smart technologies, as a rule, uses the wireless standards of communication channels IEEE 802.11X, IEEE 802.15.4, IEEE 802.16, which use only authentication protocols and privacy mechanisms that are formed on the basis of symmetric algorithms. In the conditions of the post-quantum period (the appearance of a full-scale quantum computer), the stability of such algorithms is questioned. Such systems, as a rule, are formed on the basis of the synthesis of socio-cyber-physical systems and cloud technologies, which simplifies the implementation of Advanced Persistent Threat attacks, both on the internal loop of control systems and on the external one.

The proposed creation of multi-circuit information protection systems allows for an objective assessment of the flow state of the system as a whole and the formation of preventive measures against cyber threats.

In the third chapter, models of probable threats and information protection in public networks are proposed. The most general model of the formal description of the protection system is the model of the security system with full overlap, in which a complete list of protection objects and threats to information is determined, and means of ensuring security are determined from the point of view of their effectiveness and contribution to ensuring the security of the entire tele-

communications system. It is also shown that the combination of four models (M1, M2, M3, M4) in various variants provides wide opportunities for modeling various known types of threats and their implementation. However, in connection with the continuity of the process of developing new and improving existing methods and means of implementing threats, it is necessary to use such approaches to ensuring information protection that allow detecting and preventing threats of unknown types and carrying out dynamic correction of protection behavior, adapting it to specific application conditions. The M5 basic model is described, which enables continuous refinement of threat classes and response measures, and continuous training of the adaptive component of the CSI, which, in turn, detects and prevents threats of unknown types. The M6 basic model is introduced with the aim of obtaining higher security due to the presence of a special module of internal diagnostics that diagnoses the entire protection system, decides on the correction of the SHI behavior algorithm, and makes it possible to achieve SHI fault tolerance; a special module that diagnoses the communication channel with subsequent changes in the level of protection, allows to achieve the adaptability of the SHI.

The fourth chapter is deal with the development of cryptographic primitives based on cellular automata. The definition of a cellular automaton is given and the elementary rules of intercellular interaction are described.

A number of generators of pseudorandom binary sequences have been developed based on a combination of elementary rules of intercellular interaction, as well as cell interaction according to a rule of our own development.

In the "cryptographic sponge" architecture, a cryptographic hashing function with a shuffling function based on cellular automata was developed and its statistical characteristics and avalanche effect were investigated.

A block cipher in the SP-network architecture is constructed, in which cellular automata are used to deploy the key, and the encryption process is based on elementary procedures of replacement and permutation. Substitution blocks are used from the well-known AES cipher, a description of a stream cipher is given, where a personal computer keyboard and mouse are used as the initial entropy. Random data received from the specified devices is processed by a proprietary hashing function based on a "cryptographic sponge". All developed cryptographic functions and primitives demonstrated good statistical characteristics and avalanche properties.

The fifth chapter proposes a methodology for analyzing the quality of the mechanism for validating the identified vulnerabilities of a corporate network, which is based on integral equations that take into account the quantitative characteristics of the vulnerability validation mechanism under study at a certain point in time. This technique allows to build the laws of distribution of quality indicators of the vulnerability validation process and quantify the quality of the mechanism for validating detected vulnerabilities, which allows to monitor and control the validation of identified vulnerabilities in real time during active security analysis. A method is proposed for constructing a fuzzy knowledge base for making decisions when validating vulnerabilities of software and hardware platforms with an active analysis of the security of a target corporate network based on the

use of fuzzy logic, which makes it possible to provide reliable information about the quality of the mechanism for validating vulnerabilities indirectly. The constructed knowledge base allows to form decisive decision-making rules for the implementation of a particular attacking action, which allows to develop expert systems to automate the decision-making process when validating the identified vulnerabilities of target information systems and networks. An improved method of automatic active security analysis is proposed, which, based on the synthesis of the proposed models, techniques and methods, allows, in contrast to the existing ones, to abstract from the conditions of dynamic changes in the environment, i.e. constant development of information technologies, which leads to an increase in the number of vulnerabilities and corresponding attack vectors, as well as an increase in ready-to-use exploits of vulnerabilities and their availability, and take into account only the quality parameters of the vulnerability validation process itself.

KEYWORDS

Cybersecurity, models of the threat, crypto-code constructions, simulation modelling, automation, radio engineering research, security measures.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

A	availability
ABS	automated banking system
Aff	affiliation
AFIS	automatic fingerprint identification
Au	authenticity
C	confidentiality
CCC McEliece	crypto code constructs McEliece/Niederreiter
CI	critical infrastructure
CIFS	critical infrastructure facilities systems
CIO	critical infrastructure objects
CPS	cyber-physical systems
CPSS	cyberphysical social system
CS	cybersecurity
DCS	distributed control systems
DDoS	denial of service attack
GIS	geospatial information systems
I	integrity
ICS	information and communication networks
IoTS	internet of things systems
IR	information resources
IS	information security
ISS	information security system
LDPC	low-density parity-check codes
LSI	a latent semantic indexing method
MCC	Matthew correlation coefficient
MCMC	method of Markov chain Monte Carlo
PLC	programmable logic controllers
SCADA	supervisory control and data acquisition
SI	information security

CIRCLE OF READERS AND SCOPE OF APPLICATION

Methodology for Cooperative Conflict Interaction Modeling of Security System Agents is proposed. The concept of modeling the structure and functioning of the security system of critical infrastructure facilities is demonstrated for development of a model for the implementation of a terrorist act and the degree of security of the cyber system of a critical infrastructure object. Lotka-Volterra model are used for assessing the level of security of critical infrastructure facilities. The method for assessing forecast of social impact in regional communities as a case of socio-cyber-physical systems security concept is presented.

Methodological aspects of providing information security of an individual, society and state in social networking services are investigated. Identification of threats to the information security of the state in the text content of social networking services is used for information security profiles of actors in social networking services, their classification, and information-psychological influence on actors and approaches to its evaluation. The model of conflictual interaction of civic movements in social networking services.

Counteracting the strategic manipulation of public opinion in decision-making by actors of social networking services based on the conceptual model for managed self-organization in social networking services are developed.

The problems of physical access to critical infrastructure based on analysis of biometric protection systems as a class of authentication systems are introduces. Algorithms for thinning the critical infrastructure identification system and their software are implemented.

For teachers, scientific and engineering staff in the field of cybersecurity, information technology, social engineering, communication systems, computer technology, automated control systems and economic information security, as well as for adjuncts, graduate students and senior students of relevant specialties.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

1 INTRODUCTION

A feature of the present time is the transition from an industrial society to an informational one. At the same time, information becomes a more important resource than material or energy resources. The rapid growth of computing resources, the emergence of a full-scale quantum computer increases the requirements for security systems not only for information and communication systems, but also for cyber-physical systems and technologies.

The main types of models used in modeling the behavior of intelligent agents are discussed in Chapter 2. The joint use of models of different classes together with the consideration of various aspects of the behavior of agents in conditions of cyber conflict makes it possible to obtain a synergistic effect of the proposed modeling methodology. The originality of the approach associated with the introduction into consideration of the concept of the contour of business processes as an integral object to be protected. Joint consideration of the contour of business processes of the organizational and technological system and the contour of business processes of the cybersecurity system can be considered as another condition for the manifestation of the synergy of the processes under consideration. Also worthy of attention is the idea of the spatio-temporal structure of the model basis proposed by the authors, which reflects not only the distribution of the set of models over the corresponding levels of the proposed methodology, but also sets the sequence of their interaction. This approach can be considered as the closure of a set of conditions for the manifestation of the synergistic properties of the proposed methodology for modeling conflict-cooperative interaction between the parties to a cyber conflict.

In Chapter 3 the concept of ensuring the protection of information in social networks is proposed, based on mathematical models of information protection, taking into account the specific parameters of the social network, external influences carried out on the network, taking into account the nonlinear relationships of the parameters with the protection system and the parameters of the impact of individual characteristics of users and the nature of connections between them.

In Chapter 4, practical aspects of the methodology for constructing post-quantum algorithms for asymmetric McEliece and Niederreiter cryptosystems on algebraic codes (elliptic and modified elliptic codes), their mathematical models and practical algorithms are considered. Hybrid crypto-code constructions of McEliece and Niederreiter on defective codes are proposed. They can significantly reduce the energy costs for implementation, while ensuring the required level of cryptographic strength of the system as a whole. The concept of security of corporate information and educational systems based on the construction of an adaptive information security system is proposed.

The material of the monograph is scientifically new and, in many respects, contains its own results of scientific research obtained by the authors and published in a number of scientific articles. The material is presented at a high scientific and, at the same time, accessible level, and is properly formatted.

2 METHODOLOGICAL FOUNDATIONS FOR MANAGING THE SECURITY OF SOCIO-CYBER-PHYSICAL SYSTEMS

ABSTRACT

The development of technologies and computing resources not only expanded the spectrum of digital services in all areas of human activity, but also defined the spectrum of targeted cyber attacks. The object of the study is the process of ensuring the safety of critical business processes that ensure the continuity of production and / or functioning of the company / organization / enterprise as a whole. Targeted attacks are aimed at destroying not only the business structure, but also its individual components that determine critical business processes. Continuity of such business processes is a critical component of any company, organization or enterprise of any form of government, which critically affects the earning of profits or the organization of production processes. The proposed concept of determining the security level of critical business processes is based on the need to use multi-circuit information protection systems. This allows to ensure the continuity of critical business processes through a timely objective assessment of the level of security and the timely formation of preventive measures. This approach is based on the proposed rules for determining the achievement of a given level of security, which are based on assessments of the integrity, availability and confidentiality of information arrays, as well as computer equipment in relation to various points of the organization's business processes. The use of threat integration on the internal and external contours of the protection system allows to ensure the necessary level of security and continuity of the production / technological process of critical business processes. The issues of applying the methods of situational management to ensure the safe functioning of objects of socio-cyber-physical systems are considered. Logical-transformational rules that form the foundation for building a situational type cybersecurity control system are considered. One of the main tasks of systems of this type is described – the task of replenishing the description of the situation. The use of pseudophysical logics is proposed. Various types of pseudophysical logics, the method of their construction and their interrelation are considered. Particular attention is paid to causal pseudophysical logic, as the least developed for the purposes of ensuring cybersecurity. The proposed method of providing basic security services: confidentiality, integrity and authenticity based on crypto-code constructions takes into account the level of secrecy of information transmitted over wireless channels and / or stored in databases of socio-cyber-physical systems. The use of post-quantum algorithms – McEliece / Niederreiter crypto-code constructions on elliptic / modified elliptic / lossy / low-density parity-check code provides the necessary level of stability in the post-quantum cryptoperiod (crypto-stability at the level of 10^{25}–10^{35} group operations), speed and probability of information (Perr not lower than 10^{-9}–10^{-12}). The proposed method of information exchange using wireless communication channels ensures their practical implementation on resource-limited devices (creating of CCC on the GF field (2^4–2^6).

KEYWORDS

Security level, business process, information asset, security services, multi-circuit protection systems, pseudophysical logics, description of the situation, crypto-code constructions of McEliece and Niederreiter, smart technologies.

2.1 PSEUDO-PHYSICAL LOGICS IN CONTROL OF CYBER SECURITY SYSTEMS

In [1], based on the analysis of modern control systems, it was shown that they are mainly focused on controlling objects of a physical nature. The peculiarities of socio-cyber-physical systems dictate the need to revise traditional management methods and transform the management system in such a way that it takes into account the presence of a person both in the object and in the control loop. As an approach, the use of situational management mechanisms is proposed. The comparison demonstrated wider possibilities and scope for managing socio-cyber-physical systems in comparison with traditional control systems and Situational Awareness systems. It was stated that the application of methods and mechanisms of situational cybersecurity management of these systems will require consideration of a wider class of types of relationships that exist in socio-cyber-physical systems, which should be based on the use of pseudo-physical logics. Situational management, as an approach based on management by precedents (situations), leads to the need to solve a number of problems, including not only the introduction of a clear definition of the concept of a situation, but also the creation of a classifier of situations (primarily emerging in cyberspace), instead of traditional classifiers of threats and attackers.

When working with objects of socio-cyberphysical systems, a problem arises related to the fact that in order to manage such objects, it is necessary to describe a unique object of management and take into account in this description not only its specific structure and functioning, but also people's behavior, as well as the possibilities of the object's evolution in time [2–3]. The basis of such a description can be the concept of a situation, which is more general than the description of a state in traditional control systems.

Unfortunately, even in the fundamental work [4], the definition of the situation is given in the most general form, with a distinction between the current situation at the control object (the totality of all information about the structure of the control object and its functioning at a given time) and the complete situation, as a set consisting of from the current situation, knowledge about the state of the control system at the moment and knowledge about the control technology.

Let's denote complete situations by S_i (i is the distinguishing number of the situation), and current situations by Q_j (j is the distinguishing number of the situation). Let the control system have n different ways of influencing the control object (one-step solutions). Each such decision will be denoted as U_k (k is the distinctive index of the action). The elementary act of management is as follows. If the situation Q_j has developed at the control object and the state of the control system

and the technological control scheme determined by S_i allow the use of the influence U_k, then it is applied, and the current situation Q_j turns into a new situation Q_j. The formal record of this action can be presented in the following form:

$$S_l : Q_j \xrightarrow[U_k]{} Q_l.$$

Such transformation rules are called logical transformation rules (LTR) or correlation rules. The full list of LTRs reflects the capabilities of the control system to ensure the safety of the facility. In other words, logic-transformational rules can be considered as actions of the security system that lead to a change in the situation at the control object (preventing cyber-attacks, eliminating the consequences of successful attacks, increasing the efficiency of intrusion detection systems, etc.).

Within the framework of this approach, the tasks of concretizing the description of the situation and developing mechanisms for replenishing the description of situations at objects of socio-cyber-physical systems arise from the point of view of ensuring cybersecurity. The last task can be considered as information enrichment due to the information stored in the system memory. Let's call such a problem *completion of the description*.

The main task is to build a special system that provides replenishment of the description coming to the input. In the most general form, the system can be defined as *a system of productions* of the form

$$\gamma;\alpha \Rightarrow \beta;\delta.$$

Here γ means some condition, the fulfillment of which allows the use of the product. Let α denotes a fragment of the structure (description), which is subject to transformation. The transformation itself consists in replacing fragment a with a new fragment β. Within the framework of solving the problem of replenishment of descriptions, the fragment β must be in some sense richer than the fragment α. Finally, δ is some condition modifier γ. After applying the production, this modifier changes the condition of applicability of this rule or leaves it unchanged.

Different systems of description completion differ from each other in how the products are organized and what the strategy looks like for applying them to the original description and the intermediate descriptions resulting from the completion process. A system of productions, in particular, can form a certain logical system. Such a system should reflect the patterns inherent in a given problem area and methods for constructing solutions based on a description of situations in it. At the same time, productions can be divided into three types: *deductive*, *inductive* and *traductive*. In productions of the first type, the fact β is a particular fact following from the fulfillment of the condition γ and the simultaneous presence of the fact α in the description being transformed. For inductive productions, the fact β is more general than the fact α that satisfies the condition γ. Finally, under traductive production, the facts β and α have the same degree of generality.

Production systems have a number of properties that make them a very convenient tool for describing a description completion system and its software implementation:

– autonomy – any product can be removed from it or added to it, while all other products remain unchanged, which makes the product system flexible and easily adaptable to any changes in the problem area. Adaptation of products can also occur due to changes in g and d, which allows, while keeping products in the system, to change its action. Each production is some complete piece of information about the problem area, and in this sense it does not depend on other productions. If necessary, it is possible to establish links between individual products through a system of mutual references;

– asynchrony – productions are naturally given the possibility of parallel processing. Each of them can be performed independently of the others. Various options for completing the description and special procedures that allow only such parallelisms that do not lead to ambiguous results;

– mapping to operator systems – production systems are easily mapped into operator systems of programming languages, or special languages are developed for these purposes [5]. And some programming languages have productions as their main operators.

Models in the form of production systems cover a wide class of different generative models, which includes such well-known models as formal grammars, propositional and predicate calculi, network models, and many others [6, 7].

Most often, network models are used for replenishment, which have recently received the name of scenarios [8]. A scenario can be represented by a network whose vertices correspond to facts, and to the arcs correspond to links describing relations of a special type. These relations have the property that if there are a set of paths $\pi_1, \pi_2, ..., \pi_n$ between vertices *x* and *y* and there are both facts a and b corresponding to vertices *x* and *y*, then at least a set of facts corresponding to vertices on one of the paths connecting *x* and *y*. Examples of relationships that have this property can be relationships of the type: cause-effect, part-subpart, goal-subgoal, etc. In scenarios, arcs can also characterize not a cause-and-effect relationship, but an operational relationship. The order of realization of states characterized by vertices is the order in time.

Completion of the description with the help of a scenario based on the whole-part relation can concern not only the position in space and time. Additional rules are rules of production type. Their applicability in one case or another is associated with the fulfillment of the conditions of product applicability. The condition of product applicability also includes the necessary connection between objects, which is fixed in the scenario. Thus, let's come to the conclusion that two ways of replenishing descriptions are possible: due to the formal properties of the relations used in scenarios and due to the semantics of the latter. It should be noted that the relations "cause-effect" and "ordinal relation" are transitive. And this means that without taking into account the semantics of the facts a_i, the presence of fragments (a_1ra_2) and (a_2ra_3) in the initial description makes it possible to supplement the description with a fragment (a_1ra_3), where *r* is a relation of an arbitrary type. Of greater interest may be rules that take into account the semantics of the relations themselves and the semantics of the situation in which the replenishment is formed.

By themselves, scripts do not yet solve the problem of replenishing descriptions. In fact, to solve the replenishment problem, it is necessary to create production systems in which the rules take into account the semantics of the relations included in the replenished description.

For certain classes of relations, it would be desirable to build deductive production systems that would allow the necessary completion of descriptions without storing a large number of scenarios in the system memory. One of the classes of such deductive systems are pseudophysical logics.

The name "pseudophysical logics" reflects the fact that their inference rules use the properties of human perception of the surrounding world, which has a number of subjective features. Therefore, they do not describe the objective physical world, but its subjective perception by a person, which is extremely important for cybersecurity systems.

Unlike formal systems, pseudophysical logics have a number of important features. The main ones are presented in **Table 2.1**.

Table 2.1 Features of pseudophysical logics

Logic	Feature
Pseudophysical logics are the logic of relations	Relationships play the role of variables. Therefore, pseudophysical logics are classified depending on the types of relations used. The logic of time studies the relationship of temporal relations, the logic of space – spatial, the logic of actions – relations of the type subject – action or action – place, causal logic – the relationship of relations of the type cause – effect, frequency logic – relations of the type repetition – frequency, etc. Objects, connected by relations, appear in these logics only as an invariable part of descriptions
Pseudophysical logics are logics on scales	There are two types of scales: metric and topological. Metric scales, in turn, are divided into absolute and relative. Topological scales set between the facts projected on them, relations of non-strict order, or fuzzy relations. Topological scales are closely related to fuzzy verbal assessments that are actively used by a person to describe situations (including technological ones, for example). These verbal assessments determine only a certain order of the facts on the scales. The difference in scales also determines the difference in logics, which can be metric and topological
The scales are both facts and a inference	Not only the facts are located on the scales, but the conclusion itself. The construction of production rules must take into account the orderliness inherent in reasoning within the framework of pseudophysical logics
The axioms of pseudophysical logics are based on the perception of the world by man	Pseudophysical logics contain as axioms some statements arising from the perception of the world by man. Axioms link relations of a different nature, which allows a person to replace one relationship with another
The relationship of pseudophysical logics	There are links between individual pseudophysical logics that allow forming a system of pseudophysical logics. Relations between temporal and spatial logics, which exist due to the physical laws of the surrounding world, can serve as examples of connections. Examples of connections of a different kind are the relationships that exist between causal (causal) logic and the logic of actions

When constructing pseudophysical logics, one should take into account three types of tasks for which they are intended:

a) replenishment of descriptions of situations entering the system's memory with the help of the knowledge that is already stored in the system about the control object, the history of control and the laws of control of the object;

b) checking the reliability of the incoming description of the situation, identifying contradictions in this description and its compatibility with the information that is already stored in the system;

c) participation in the formation of decisions on management and verification of the possibilities for implementing the selected control action.

These types of tasks are extremely relevant for cybersecurity systems (**Fig. 2.1**).

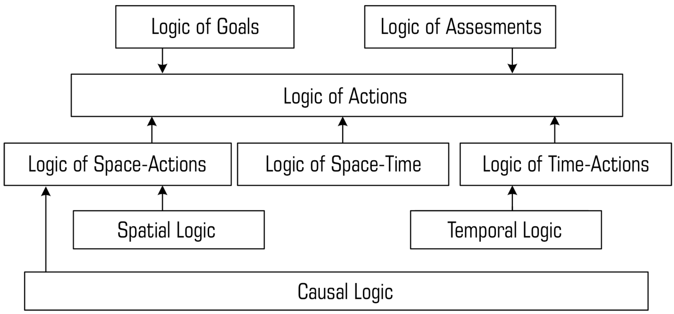


Fig. 2.1 Relationship between pseudophysical logics

Each pseudophysical logic can be considered as a system having the structure shown in **Fig. 2.2**. It shows that, first, some structure of facts or events characteristic of a given pseudophysical logic (temporal structure, spatial structure, etc.) is distinguished from the initial description. It highlights some units such as phenomena, events, processes, facts and defines the relationship between them from the considered group of relationships.

This part, in fact, does not refer to the actual pseudophysical logic and its functioning is based on procedures that perform the transition from a description in a natural language to a formal representation. The representation model reflects those basic patterns of perception that are characteristic of the control system (or a person, if its perception is imitated). For specific pseudophysical logics, this model turns into a model of time, a model of space, etc. Finally, the inference model contains rules that help to complete the description of situations.

Pseudophysical logics such as spatial logic and temporal logic have been used recently in IDS systems [5, 9]. At the same time, in spatial logic, cyberspace is primarily considered as space, the coordinates of which are IP addresses. In the temporal logic, the relationships concern both the

sequence of the attacks and the temporal characteristics of the attacks (the total duration, the times of the implementation of each of the phases of the attack, etc.).

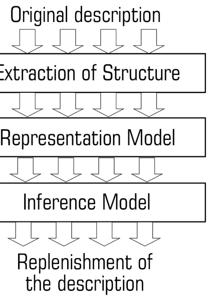


Fig. 2.2 The structure of pseudophysical logic

Regarding the logic of assessments, it can be noted that it can become the basis for the selection and development of various metrics used in cybersecurity [10].

Less developed in relation to cybersecurity systems is causal logic, which must work with relationships that connect cause and effect. If causal logic is interpreted broadly enough, then it can be extended almost to the theory of inference, which considers not only traditional deductive inference systems, but also inductive-type inferences or inferences of the "from particular to particular" (traductive) type [11].

There are at least five main types of such "cause-effect" relationships:

1. *Energy reason.* The reason for the change in w observed in some phenomenon or process Π_2 is the transfer of some energy v from the phenomenon or process Π_1. In this case, it is possible to say that v is the cause of w, and w is the effect of v.

2. *The reason for the change in w in Π_2 is not the effect v itself.* The supposed cause rather plays the role of the "last push", after which the process leading to w begins to develop in Π_2. However, in this case, it is possible to also consider v the cause (original cause) of w, and w the consequence of v.

3. *The reason for the change w is the information v, which contains an indication for Π_2 about the nature of the required change.* It is possible to assume that between this indication of v and the change in w there is a cause-and-effect relationship.

4. *The change in w occurs due to the fact that in Π_2 there are two subprocesses, or systems that interact with each other and generate the change in w itself.* Such causes of change may also arise as secondary after the appearance of the primary causes indicated in the preceding paragraphs. Let's note that the interaction within the process can determine the development of the process itself, which is expressed in the chain of its changes $w_1, w_2, \ldots, w_n$.

5. *Some "fundamental" law can act as a reason.* Some "fundamental" law can serve as a reason, according to which every process tends to some stable states (for example, the existence of a certain ratio between the number of successful and repulsed attacks).

The relationship of causes and effects can be very diverse and not always obvious.

From this point of view, it is possible to give the following classification of cause-and-effect relationships (**Fig. 2.3**).

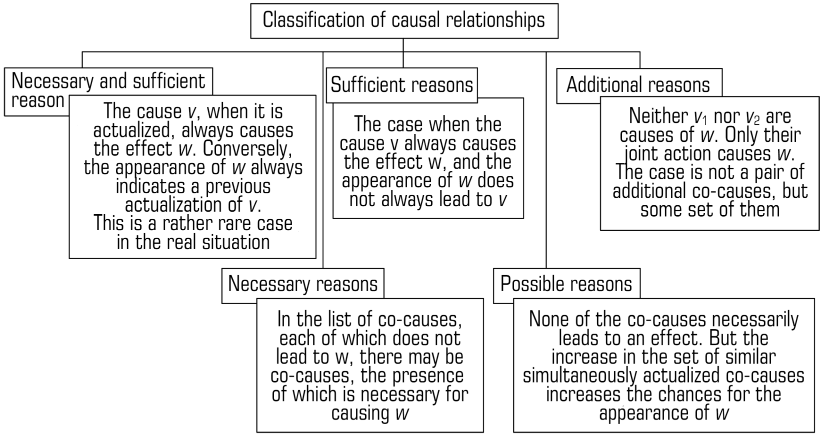


Fig. 2.3 Classification of causal relationships

The main property of cause-and-effect relationships is their antisymmetry and transitivity, which allows for any phenomenon that can act as a consequence to build a tree of causes or co-causes that can give rise to it [12–14].

This is where the temporal and causal logics meet. We have already said that reasoning about the future is modal. In the future, not one sequence of events may be realized, but a whole fan of such sequences. And when deriving from causes to effects, this must be taken into account. Therefore, causal scenarios need to be supplied with additional information (weights) about time delays in the onset of effects after immediate causes. These weights can be used to label the arcs shown in the scenario, which makes it possible to draw richer conclusions in causal logic.

Achieving the goals of their business by companies is possible only with the effective use of information technology. The downside of this use is increased vulnerability to cybersecurity threats. Vulnerability identification and risk assessment strongly require an information security risk assessment. The data used for identification procedures is in most cases uncertain, which makes it a challenge to identify risks and vulnerabilities. So-called "vulnerability identification errors" can occur if false positive vulnerabilities are discovered or if vulnerabilities remain unidentified (false negative).

“Clear identification” in this context means that all identified vulnerabilities do pose a security risk to the organization.

2.2 CONCEPT OF DETERMINING THE LEVEL OF SECURITY

In order to identify vulnerabilities in the information security (IS) risk assessment, security experts analyze the organization’s assets. Due to the fact that the probabilities, consequences, and losses of vulnerabilities cannot be accurately determined [15], methods such as brainstorming, checklists, scenario analysis, impact analysis, and cause analysis are used to identify vulnerabilities [16]. These methods use undefined input to identify a vulnerability. However, it should be noted that business security needs are not properly considered; security checklists and standards used to identify vulnerabilities do not take into account company-specific security requirements [17]. Further, increasing uncertainty is the intentional behavior of an attacker when exploiting vulnerabilities for malicious purposes. This is explained by the fact that predicting human behavior is associated more with existing vulnerabilities and their consequences [18], rather than with preparation for future attacks. As a result, modern approaches identify risks and vulnerabilities under conditions of a high degree of uncertainty, which can lead to errors [3, 19–22].

Practice shows that today it is possible to clearly distinguish two main groups of methods for assessing security risks [23–27]. The first group of methods allows to set the risk level by assessing the degree of compliance with a certain set of information security requirements. The second group of information security risk assessment methods is based on determining the probability of attacks, as well as the levels of their damage. In this case, the risk value is calculated separately for each threat and, in the general case, is presented as the product of the probability of a threat being realized by the amount of potential damage from this threat. The value of the damage is determined by the owner of the information, and the probability of the threat being realized is calculated by a group of experts conducting the audit procedure.

A distinctive feature of the methods of the first and second groups is the use of different scales to determine the magnitude of the risk. In the first case, the risk and all its parameters are expressed in numerical, that is, quantitative values. In the second case, qualitative scales are used.

The use of the results of a quantitative risk assessment in the formation of an information security system (ISS) is due to several reasons. Firstly, quantitative risk assessment allows to compare the benefits and costs of implementing GIS, thereby determining the effectiveness of investments in information security (ISec). Secondly, many currently widely used standards in the field of information security and information technology (IT) are based on a risk-based approach. It is also worth noting that there is a successful risk management practice in other areas, such as economics and finance, politics, ecology, production, and industrial safety. This allows to integrate risk management processes in certain areas into a single enterprise risk management system. One of the main problems of existing approaches is the difficulty in obtaining objective quantitative

assessments of IS risks, which require a large amount of initial data. Predicting individual risk parameters with acceptable accuracy is a very laborious task, and it is difficult to obtain an accurate quantitative estimate.

A significant influence on the formation of a list of critical business processes, which makes it difficult to create security systems, is exerted by the use of various technologies and elements within the framework of the integration and hybridity of technologies of socio-cyber-physical systems. Often, when assessing the risks that arise during the operation of an information system, causal relationships between identified risks are not taken into account. An information system (IS) is understood as "a set of information contained in databases and information technologies and technical means that ensure its processing". Based on the definition and analysis of cyber-physical systems, the following types of components of modern hybrid / complex IS can be distinguished: information assets (IA), software (SW), hardware (TS) and communication lines (CL). A structural diagram of the types of IS components is shown in **Fig. 2.4**.

Therefore, the set of IS components can also be represented as:

$$IS = \left\{ IA, SW, HW, CC \right\},$$

where *IA* is the set of information assets, *SW* is the set of software; *HW* is the set of hardware; *CC* is a set of communication channels.

A destructive state is understood as an undesirable and unplanned state of an IS component in which it finds itself as a result of the implementation of one or more threats. During the analysis of various regulatory documents on information security, the theory of reliability and a survey of specialists in the field of IT and information security, the main destructive states were identified for each type of IS components:

1) information asset (IA):
– unavailable (accessibility violated);
– compromised (violated confidentiality);
– changed (integrity is broken);
2) software (SW):
– unavailable (failure occurred);
– hacked (unauthorized access (UA) obtained by an attacker or user privileges increased);
– changed (unauthorized change of code and / or configuration);
3) technical tool (HW):
– unavailable (a temporary failure has occurred);
– inoperable (a failure has occurred requiring repair or replacement);
– lost (there was a loss or theft from the rightful owner);
4) communication channels (CC):
– unavailable (failure or failure has occurred);
– hacked (acquired UA by an attacker).

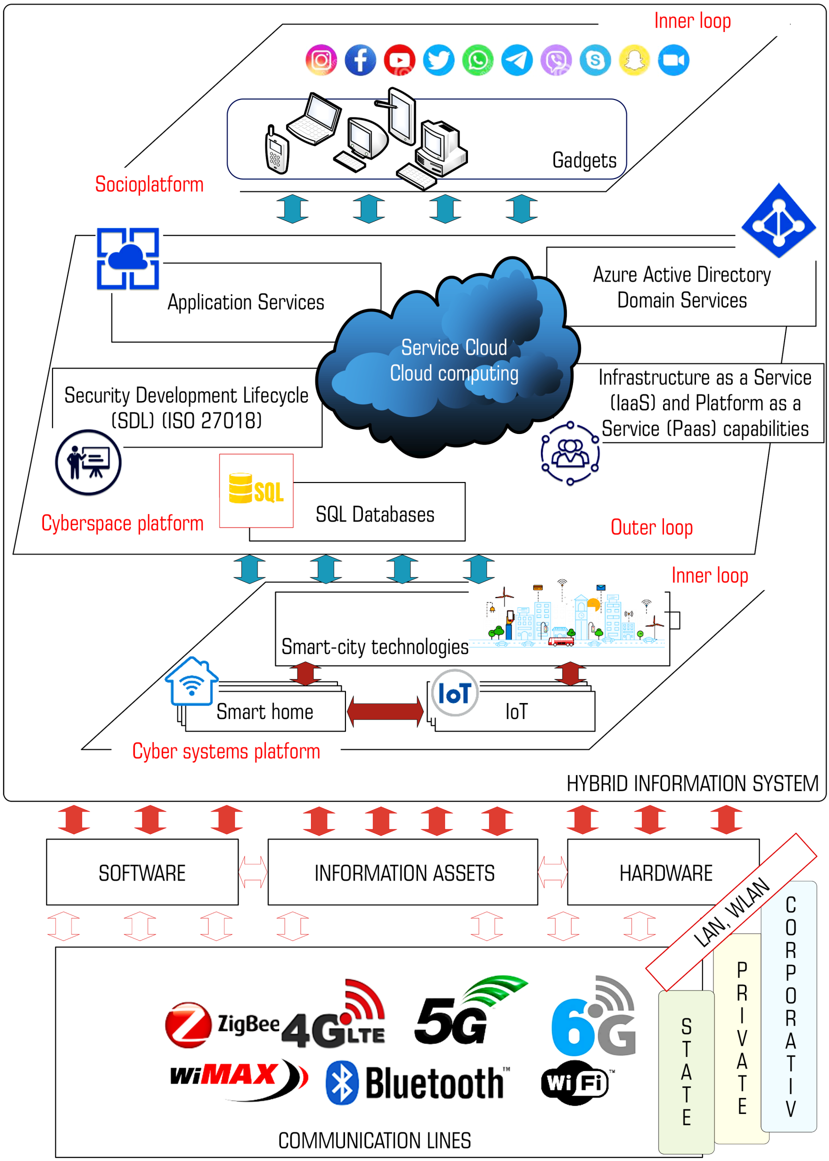


Fig. 2.4 Structural diagram of the types of components of the information system

A transition is understood as a change in the state of an IS component from normal to destructive as a result of a threat. The reasons for the transition of the IS component to a destructive state can be:

– the impact of the source of threats on the IS component;

– the completed transition of the associated IS component to the destructive state.

The source of threats is understood as the subject of access, a material object or a physical phenomenon that causes a threat to information security. The set of threat sources includes sources of four types:

$$ST = \{ND, TS, UV, IV\},$$

where *ST* are sources of threats, *ND* are natural and man-made disasters; *TS* are technical means and systems; *UV* are unintentional violators; *IV* are intentional violators (intruders).

The study is based on information security models that describe the concepts used (e.g. assets, vulnerabilities and security requirements) in managing and assessing information security risks. The subject of the research is the development of the Concept for determining the level of security of critical business processes in the context of modern mixed cyber threats. The object of research is the process of ensuring the security of critical business processes. Asset-related concepts describe critical assets and their security, while risk-treatment-related concepts describe security solutions, requirements, and security features used to mitigate risks. The main trends and approaches to determining the level of security are demonstrated in **Fig. 2.5**.

To build an integral security system for hybrid information systems, it is not enough to use only the principles that regulate in international regulators. An integrated approach is needed not only for the analysis of information assets, but also for the definition of critical (continuous) business processes that ensure the achievement of the goals of the company and / or organization. This approach requires the consideration of new approaches based on the integration of known methods and methods for assessing risks and computer vulnerabilities, taking into account the synergy and hybridity of targeted threats to elements of the IS infrastructure. In addition, it is necessary to form new requirements for assessing the security level of hybrid ISs, which are not only logically but also physically separated in space, use different technologies, and form both cyber-physical and socio-cyber-physical systems. This approach to building cyber-physical systems requires building security systems for each of the circuits / systems. This creates the need for multi-loop security systems with an integrated approach that takes into account both individual threats to the loops (internal and external) and their synergy for the attacker to build mixed (targeted) attacks [21, 22]. As a rule, when assessing security risks, after identifying the assets, the threats that may arise are determined. However, there are problems associated with the fact that it is impossible to determine whether the lists of threats, vulnerabilities or security controls used are complete and comprehensive.

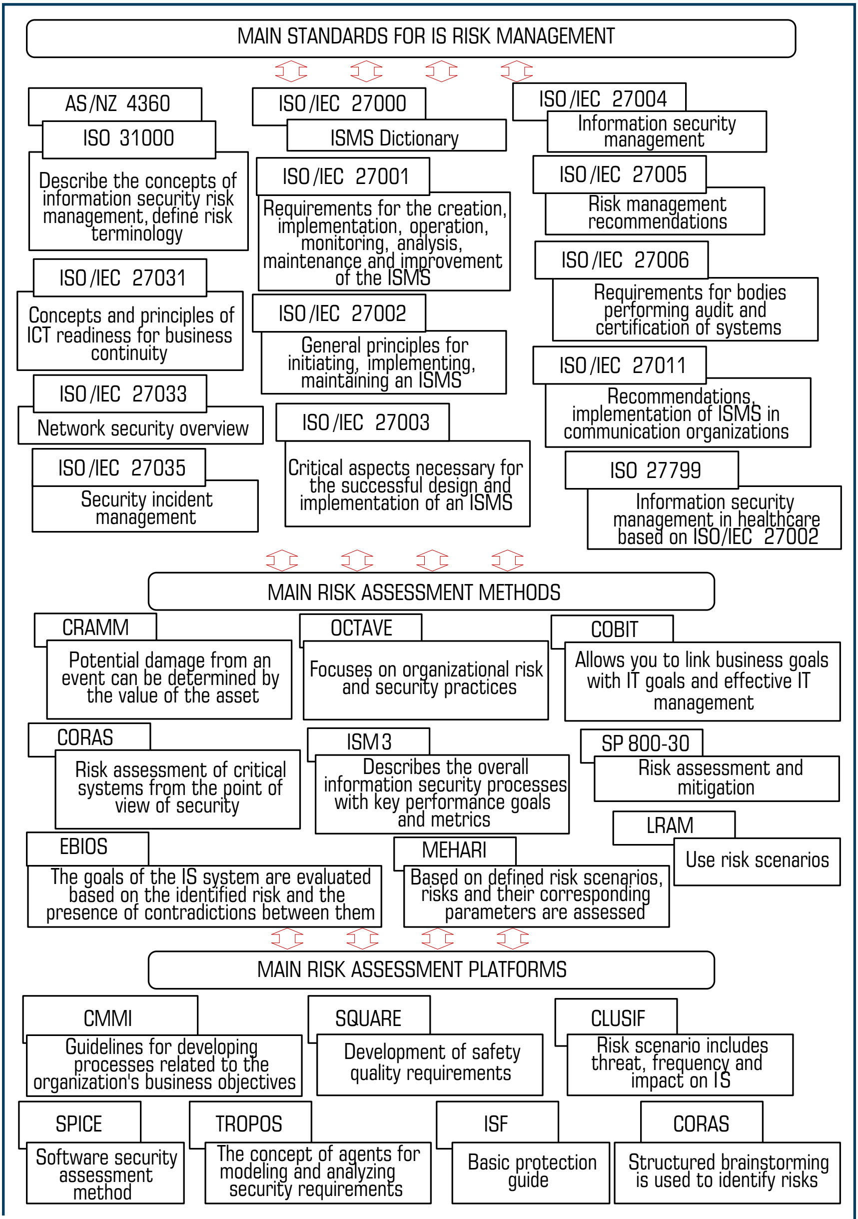


Fig. 2.5 Basic approaches to information security risk management

When determining "real" risks for a company, there are many uncertainties in assessing probabilities as risk values. To illustrate the challenges of assessing risk in an uncertain environment, consider calculating the likelihood of a hypothetical encryption vulnerability on a web server. Let the web server need to determine the probabilities of the following outcomes:

– the offender exploits an encryption vulnerability in a web server;
– a hacker or criminal exploits an encryption vulnerability on a web server;
– a hacker or criminal will not be able to exploit any vulnerability.

More data is needed to determine the likelihood of detected web server threats and the parameters associated with this scenario need to be taken into account. The following data should be identified in the probability assessment and assessed for their availability:

– the number of known exploits for the web server version;
– the number of unprotected exploits for the web server version;
– criticality of exploits;
– the level of detection of all vulnerabilities by an attacker;
– coefficient of successful use;
– number of users of the web server application;
– the ratio of friendly and malicious users accessing the web server;
– impact on controls.

From the data that is needed to determine the probabilities, it is possible to form a diagram of dependencies between the parameters (**Fig. 2.6**), which can be used to estimate the probability.

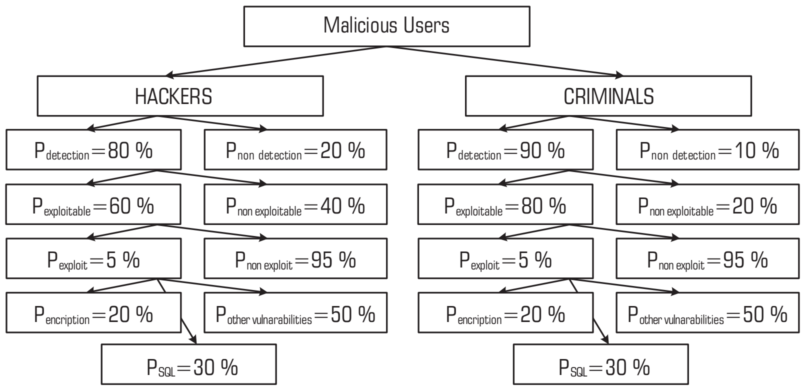


Fig. 2.6 Dependency tree with the probabilities of the implementation of threats

Analysis of **Fig. 2.6** showed that attackers accessing a website fall into two groups – hackers and criminals. The sublevels below hackers and criminals are the same. Each was assigned

a Vulnerability Detection Rate, meaning the likelihood that they successfully discovered vulnerabilities. The following is the ratio of exploited vulnerabilities. Only a few vulnerabilities can be exploited because they have not been fixed. The next level concerns whether the hacker / criminal is capable of exploiting unprotected vulnerabilities. The last level of the tree is the likelihood that this vulnerability will be exploited. Some vulnerabilities are likely to be exploited more than others. This example assumes independence of variables (for example, hacker, criminal, encryption and SQL vulnerability). However, it is often difficult to determine whether the parameters are independent. This is due to the fact that it is necessary to know:

– the intentions of the attacker (for example, what is the difference between a hacker and a criminal);

– technical details of vulnerabilities (for example, details about encryption and a problem with SQL);

– environment (e.g. administrative and technical security functions applied);

– parameters and their relationship with each other.

Estimates for each parameter of the tree, obtained as a result of expert evaluation, are presented in **Fig. 2.6**. Most of the probability values in the dependency tree are only expert estimates that do not claim to be reliable. Rather, these artificial values are used to demonstrate how individual values affect the overall likelihood, and also to provide a statement about the likelihood of a hacker and a criminal using a web server.

Before presenting the detailed results of the probability score for a hacker and a criminal, one would expect the web server to be of medium risk. The results for probabilities since the start of this analysis are as follows:

– the offender uses an encryption vulnerability on the web server, probability = 0.144 %;

– a hacker exploits an encryption vulnerability on a web server, probability = 0.384 %;

– a hacker or criminal uses an encryption vulnerability on a web server, probability = 0.528 %;

– a hacker or criminal will not exploit any vulnerability, probability = 97.36 %.

Before defining risk based on security requirements, it is necessary to compare models to see if the relationship between risk and security requirements is defined. Before the comparison, it is necessary to give the basic definitions of the elements that will be used for an information security model based on these three models using similar terminology. The main definitions of the elements are as follows.

The security goal is determined by the business requirements that are affected by the risks. Business requirements describe security needs in terms of business operations, where a business operation is a set of activities that are defined and can be modeled as a business process. A security requirement is a refinement and additional specification of a security goal and represents constraints on the functions of a system where these constraints implement one or more security goals [18]. It is necessary to enter the following definitions:

– risk treatment is the process of selecting and implementing security functions to modify risk based on security requirements;

– security function – security requirements in the form of administrative, physical or technical controls and are applied to an asset in order to comply with a security requirement;

– assurance is an assessment of the security function and is used to determine whether the security requirements are met;

– assurance – assurance that security features reduce risks to assets and that assets are protected as required;

– an asset may consist of hardware, software, information systems, or any physical means used to fulfill an organization's business requirements. An information asset is a refinement of an asset that is made up of data;

– risk is a combination of a likely event and its impact, which can lead to a violation of safety objectives;

– an impact is an adverse change in an event that violates the safety objectives of an asset;

– an event is a threat that exploits a form of vulnerability;

– a vulnerability is a weakness in an asset or control associated with a security objective;

– a threat is a potential attack or incident that could lead to an adverse impact on an asset;

– the business process model is a detailed description of the business process, including the activities, agents, artifacts, and roles involved in the modeling notation;

– an artifact is a product that was created or changed as a result of a technological action;

– a role is a set of actions that has been assigned to the participants in the process to determine the functional responsibility.

A proposed method applies existing models such as information assets and security requirements to business process models (BPM). Business process models describe the actions of a process in an organization to achieve a goal. From BPM, information assets, participants, and computer system facilities for risk assessment can first be determined. The criticality of each information asset can be defined by business process objectives in the form of security objectives. After that, the security requirements specifying the security objectives can be identified and used to argue for the correctness of the procedure for correctly identifying vulnerabilities. Security features related to business process activities that use an information asset are compared with security requirements to identify vulnerabilities. The proposed approach provides an assessment of security requirements within business process models to identify vulnerabilities, eliminate identification errors (false positives, false negatives and true positives) of vulnerabilities, regardless of the business processes used. However, it should be noted that not all vulnerability identification errors can be eliminated by applying this approach. The difference between the proposed method and existing approaches lies in the explicit assessment of security requirements in a business context (in business process models) to accurately identify vulnerabilities. Eliminating vulnerability identification errors will reduce the amount of money spent on the implementation of ineffective security measures, which is the result of meeting business security requirements.

The concept of determining the level of security is based on an extended model of information security (**Fig. 2.7**), based on the models considered earlier and representing these relationships.

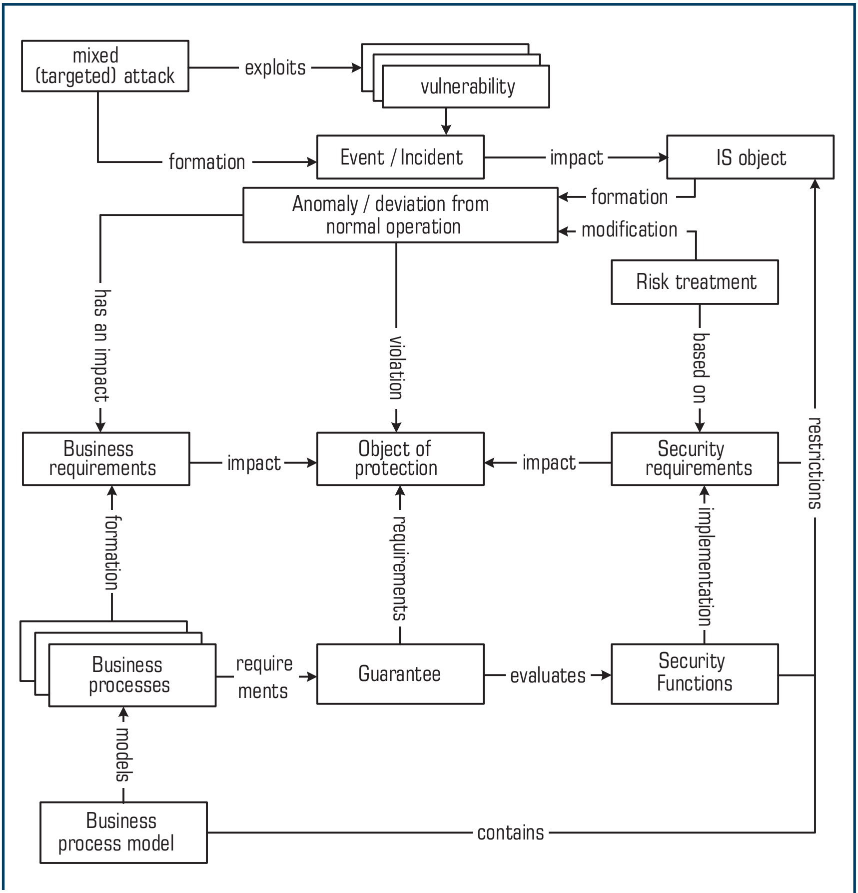


Fig. 2.7 Extended information security risk model

The need to introduce an extended information security model into consideration is caused by the fact that none of the analyzed models considers the relationship between risk, security requirements, security controls and assets. This extended information security model proposes to consider risk in terms of security requirements because it provides a combined view of concepts related to risk, risk treatment, and security requirements.

This extended information security risk model uses model elements and relationships between elements from the previously mentioned models. Rectangles are used to represent elements, and text-annotated arrows are used to describe relationships between elements. In the model that follows, adds elements for elements such as assurances, business requirements, and business

process modeling that are not present in existing models. Labeled arrows between elements such as vulnerability and risk are added to describe the context of the relationship and to clarify the relationship between concepts related to risk, risk treatment, assets, and security requirements. Labels are used to describe relationships between elements.

Compared to existing models, this extended information security risk model provides a combined representation of concepts related to risk, risk treatment, asset requirements, and security, similar to models [28–31]. The difference is that risk treatment and asset-related concepts are linked through risk and security requirements, and not just through risk itself. Model elements such as risk, vulnerability, security objective, security requirements, security controls, and asset are linked together, showing that risks and vulnerabilities affect security objectives, security requirements, and assets. Models [28, 29] do link risks to security objectives and requirements, but they use this relationship to indicate that security requirements reduce risk and that the significance of the risk is determined by the security criterion. Models [32–34] associate threats with security requirements. This indicates a breach, but misses the link between security requirements and controls, and also lacks the concept of vulnerabilities in the model.

In the model [29], vulnerabilities and assets are linked through security policy to security requirements, but risk treatment concepts are not explicitly specified. The advanced information security model shows the impact of risks and vulnerabilities on security objectives, security requirements, and assets. That is why it can help to better understand the relationship between concepts related to risk, assets, security requirements and risk treatment. Thus, it will allow for a better integration of these concepts into existing approaches to risk assessment. In addition, it is used as a basis for determining risk in terms of security requirements. This is made possible by the relationship between risk and vulnerabilities, security objectives and associated security requirements, and not just in terms of threats and vulnerabilities.

Taking into account the remarks made, it can be argued that the extended information security risk model is the basis for determining the risk in terms of security requirements. Further, the definition of risk is developed based on the relationship between risk, vulnerabilities, goals and security requirements.

Risks and vulnerabilities violate security objectives arising from business requirements by failing to ensure the confidentiality, integrity, and availability of information. This can be detrimental to the organization. In an extended information security risk model, this is depicted as a relationship, labeled "violating," between a risk or vulnerability and a security goal. If a security requirement is not implemented, implemented incorrectly, or not followed, it will adversely affect the security goal and ultimately business requirements. This is because security requirements refine the purpose of security by defining the requirements for ensuring the confidentiality, integrity, and availability of information. Therefore, non-compliance with security requirements can be expected to be detrimental to the organization and therefore constitutes a security risk to the organization as a whole.

The link between a risk or vulnerability and a security goal is that the former can violate the security goal specified by the security requirements and implemented through the security functions,

thereby harming the organization. Therefore, risk can also be defined as "non-compliance with safety requirements that causes harm to the organization". Therefore, both a risk and a vulnerability can be identified by the deviation or non-compliance with the security requirements by the implemented security functions. Security functions are implemented by administrative, physical and technical controls that meet security requirements. This means that the correct implementation and operation of security functions in relation to compliance with security requirements is key to preventing risk to the organization, as well as to identify risks and vulnerabilities.

Business process models can be used to assess risks, vulnerabilities, and security features that describe the operation and core values of an organization. Process actions describe what process participants or agents must do. An agent can be a person or a system performing an action. According to [36], business process models describe the processes of value creation in an organization and can be considered as a place where risks materialize, information is generated and security functions are performed.

It is the business process model that contains the information assets. Within a business process, information is processed to achieve the purpose of the process. The information is used by the actors (e.g., people) and systems (e.g., application or network) of the process because such information represents a business transaction. Information assets are subject to security requirements to ensure that the purpose of the process is achieved. The security requirements of an information asset depend on the purpose of the business process, the context and significance of the information related to the company, product, service or person, and represent a constraint. The security requirements are implemented through the security functions for the information asset. Security features provide protection in terms of confidentiality, integrity, and availability of an information asset. Security features such as authentication mechanisms ensure compliance with information asset security requirements (**Fig. 2.8**).

Information assets are key because, when processed in a business context, they bind business and security goals to their requirements and it is to them that security functions are applied. The correlation between information assets, business process models, security requirements, and security functions can be used in risk assessment to identify vulnerabilities. The security requirements can be assessed using the security functions applied to the information asset. The security of an information asset can be assessed in the context of a business process by the activities of the value-creating processes that use the information. By using these elements in an assessment, it can be ensured that the required security can be implemented and that the organization is not at risk.

The proposed security breach risk assessment approach uses correlations between security requirement elements, information assets, business process models, and security functions to identify vulnerabilities.

Initially, the list of information assets is formed on the basis of business process models (1). Information assets can be identified by information used in critical business processes. Information assets are characterized by the security requirements that can be defined for them and represent constraints (2). Artifacts such as business process and security objectives, security policies, or

security best practices can be used to define security requirements. These information asset security requirements are analyzed and evaluated in the process activities of the business process model (3) against the implemented security functions applied to information assets (3) to identify vulnerabilities (**Fig. 2.8**). To determine if the information assets are appropriate for the implemented security functions, the security requirements of the asset must be evaluated in each activity of the business process model where the information assets are used.

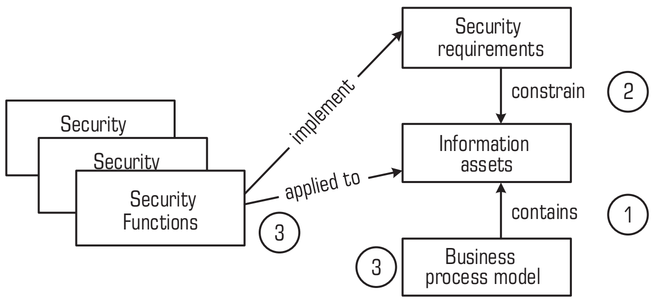


Fig. 2.8 Correlation of concepts and the order of use of concepts

The benefit of evaluating information asset security requirements using a business process model is that the basis for the assessment – the security requirements that provide true value – is defined and explicitly evaluated in the operational business context. Assertion about the security of information assets can be provided by the results of risk assessment of processes and information assets, which show only really relevant vulnerabilities. This statement can be formulated as not only vulnerabilities are defined, but also security needs and the need to fulfill them. As a result, security vulnerabilities and security operations can be identified more precisely than simply based on security best practices for any individual asset. In addition, interdependencies between information assets and / or processes can also be taken into account, since the security requirements of information assets are systematically assessed during the implementation of a business process. The difference from other approaches that use parts of business process models, information assets, or security requirements is that vulnerability identification for an information asset is based on an explicit assessment of the security requirements of the asset, taking into account the actions of business processes and the implementation of security functions.

The structure of the characteristics of security requirements that define the security needs of an information asset, taking into account the processing of information, containers that process information, as well as the processes required for the security of containers, is presented in **Table 2.2**.

Table 2.2 Security Requirements for Information Assets

Information asset	Security services	IT Security Processes
Processing		
Data	Integrity, Confidentiality and availability security objectives rating	n/a
Containers		
Primary Systems	Integrity, confidentiality and availability requirements for the systems that processes information	IT security processes that ensure the security of systems
Organization / People	Integrity, confidentiality and availability requirements for the actors or the processes that handle the information	IT security processes that ensure security in the organization
Environment / Physical	Integrity, Confidentiality and availability requirements for the environment where the information is physically available	IT security processes that ensure security for facilities and the workplace

Thus, to implement the proposed concept, it is necessary to determine not only possible mixed (targeted) cyber attacks, but also to form a set of rules for determining the achievability of the required security level. This approach ensures the integration of preventive measures for possible computer incidents / vulnerabilities, mixed (targeted) attacks, and will also automate the process of their formation, taking into account business goals. In addition, the proposed solution allows to create and take into account critical business processes, conduct modeling based on the Prolog program.

Formation of a set of rules for determining the achievability of a given security level. The set of rules for determining the achievability of a given security level is proposed to be formed as follows. First of all, it is necessary to perform the identification of vulnerabilities, which is performed as follows.

The first step assesses the degree of implementation of security functions at the processing level and their compliance with the security objectives of information assets (second step). The second step evaluates the information asset containers (i.e., systems, actors, and environment) and security requirements.

Implemented security functions are evaluated to determine conformity with the security objectives. However, before the security functions can be assessed, it is necessary to determine where in the business process information assets are created, processed and transferred. These business process points are defined as entry points (EPs), processing points (PPs), and communication points (CCs). Entry points (EPs) describe the actions by which available information is made available for processing by entering the system. Processing points (PPs) describe activities in which information is stored permanently in electronic form or modified (processed). Communication channels (CC) describe activities in which information is transferred between the activities of a process.

Information can be transferred across organizational boundaries, geographic locations, or across departments. EP, PP and CC can be identified by keywords (e.g. enter, process, save, send) of business process operation descriptions. Entry points, processing points and communication channels determine the input, storage, processing and transmission of information. Through these process points, the security objectives of processing an information asset are evaluated.

For each EP, PP and CC, the degree of implementation of security services (functions) (such as access control, authorization, data verification, communication, encryption, performance) is determined. These security functions are subject to evaluation because they are closely related to the security objectives by definition. For example, integrity concerns the protection of accuracy and completeness; therefore, access control, authorization, and data validation are checked to ensure integrity.

Confidentiality is concerned with preventing unauthorized disclosure of information, so access control, authorization, communication, and encryption are checked to ensure confidentiality.

Availability means that the assets are available and therefore the implemented contingency measures and system performance are checked to ensure availability.

All possible ratings defined for access control (AC), authorization (A), input validation (D), communication (C), and encryption (E) [34] are shown in **Table 2.3**. For each level, its rating and abbreviation are determined, for example. AC0 means access control level 0; A2 stands for authorization level 2. After defining the implementation levels of the security function, it is necessary to determine whether the security objective of the information asset is met at each of the EPs, PPs and CCs where the information asset is used.

Table 2.3 EP, PP and CC rating criteria

EP / PP measures			**CC measures**	
Access control & accountability	**Authorization (access right)**	**Data input validation**	**Communication**	**Encryption**
AC0: Unauthenticated user	A0: none	D0: None	C0: External unauthenticated partner	E0: None
AC1: internal user	A1: Read	D1: Manual	C1: External authenticated partner	E1: Weak encryption
AC2: authenticated user	A2: Execute / process	D2: Downstream validation	C2: Internal network partner	E2: Standard encryption
AC3: System user	A3: Write / update	D3: Value verification	C3: Internal authenticated partner	E3: Strong encryption
EP / PP security level	A4: Full control	D4: Value verification and completeness	CC security level	

In the next step, the security function implementation rating for each EP, PP and CC is assessed against the information asset's security objective level. The set of security functions is evaluated taking into account the action on the information and the security goal to be achieved. EPs are only evaluated for integrity through security, access control, and data validation features. PPs are evaluated based on integrity and confidentiality through security, access control, authorization, and data validation features. CCs are evaluated for integrity and confidentiality through the use of communication security and encryption features. Availability is assessed for EPs, PPs, and CCs that use the systems, based on system performance and contingency measures taken.

The evaluation of the security function implementation rating against the information asset security objective rating is supported by a predefined set of rules for checking compliance. For example, an access control score in EP1 (entry point 1) evaluated as AC0 (access control level 0) is compared to the defined rules for security goal integrity level 2. **Table 2.4** shows a fragment of a complete set of rules defined to ensure the integrity and confidentiality of security goals [38]. Scores for EP / PP and CC can be "good" (fair – requirements met), "poor" (not enough – requirements not met), "n/a" (not applicable), or "n" (unknown – not rated).

Table 2.4 Security goal rule set table (excerpt)

Security Objective		Integrity		
		Level 1	Level 2	Level 3
1	2	3	4	5
Access Control				
Unauthenticated user	AC0	EP and ≥D2 PP	EP and D4 PP and ≤A1	EP failed PP and ≤A1
Internal user	AC1	EP and ≥D1 PP	EP and ≥D2 PP and ≤A2	EP failed PP and ≤A1
Authenticated user	AC2	EP and ≥D1 PP	EP and ≥D1 PP and AS and≥D1	EP and ≥D2 PP and (A3 or A4 and D4)
System user	AC3	EP PP	EP PP	EP PP
Authorization				
None	A0	PP	PP	PP
Read	A1	PP	PP	PP
Execute / process	A2	PP	PP≥AC1	PP and ≥AC2
Write / update	A3	PP and ≥D3	PP and D4 AC2 and≥D1	PP and (AC2 and D4) AC3
Full control	A4	PP and ≥03	PP and D4 AC2 and D2	PP and (AC2 and D4) AC3

Continuation of Table 2.4

1	2	3	4	5
Data validation				
None	D0	EP failed	EP failed	EP failed
Manual	D1	EP and ≥AC1	EP and AC2	EP failed
Downstream reasonableness validation	D2	EP	EP and ≥AC1	EP and AC2
Value verification	D3	EP	EP and AC2	EP and AC2
Value verification and completeness	D4	EP	EP	EP and AC2
Communication				
External unauthenticated partner	C0	CC and ≥E1	CC failed	CC failed
External authenticated partner	C1	CC	CC and ≥E2	CC and E3
Internal network partner	C2	CC	CC	CC and ≥E2
Internal authenticated partner	C3	CC	CC	CC
Encryption				
None	E0	CC failed	CC failed	CC failed
Weak encryption	E1	CC	CC failed	CC failed
Standard encryption	E2	CC	CC	CC failed
Strong encryption	E3	CC	CC	CC

The rules can be read as follows for the first level of integrity and access control to security functions (each cell represents one or more rules):

– AC0: If an EP is rated AC0, then the Data Entry Review score must be at least D2 for that EP. PP AC0 rating is ok;

– AC1 and AC2: If the EP is rated AC1 or AC2, the data entry validation rating must be at least D1. PP rating AC1 / AC2 is ok;

– AC3: EP rating AC3 is acceptable. The PP AC3 rating is ok.

The assessment of the integrity and confidentiality of information asset security objectives is based on the same security functions and follows the same procedure. Different categories such as "performance" and "measures" are used for the accessibility of a security goal. Scores how often accessibility requirements have been met in the past, with a performance rating. Implemented continuity measures are rated with a "measure" rating. The difference from the integrity and confidentiality assessment is that only system containers are considered in the availability assessment. Rule sets are not static and can be changed as required by company policy. The rules were defined

using the knowledge of security experts and taking into account the dependencies of the security functions on the level of the security goal.

In the next step, the containers of information assets (e.g., information systems, personnel, and environment) are evaluated in terms of information asset security requirements. Security requirements for containers of information assets are evaluated at each technological operation in which information is processed. This is done in the form of EP, PP and CC using information gathering methods such as on-site interviews and document reviews. The identified EPs, PPs, and CCs are evaluated by the security assessor based on evidence that the security requirements for the system, organization, or physical environment are met. This evidence may be obtained from the system configuration, system specification, company security policy, technical documentation, or implementation examples. IT security processes are assessed through system testing, verification, and review of process performance documentation. Assessing the IT security process helps identify technical issues and ensure safe operations; it also defines an organization's ability to detect, prevent, or mitigate security problems. The security of an IT process is determined by whether problems are identified in the implementation of the business process or not.

The next step is to specify the information asset security requirements. First, an appropriate security goal rating for integrity, confidentiality, and availability must be selected based on the information asset's security needs. Second, the security requirements of the container must be specified; a description of what needs to be protected, as well as a specific implementation in the containers that process the information. The company's security policy, organizational procedures, and security best practices can be used to identify and define security requirements.

The generated set of rules can be considered as the basis for the implementation of situational management of the security system, and in particular, the business process security management system.

Dependencies between access control and authorization, as well as data validation in relation to input or processing of information, arranged in the form of a set of rules (**Table 2.4**), were implemented in Prolog to support automatic evaluation generation.

Prolog is a declarative programming language based on facts and rules. The procedures for working with them are implemented in the programming language itself and do not require programming costs. Prolog was also chosen because it is possible to generate logical search specifications to determine when a security function implementation becomes true with respect to the security goal level. This characteristic of Prolog can also be used to determine what security functions are needed (if not known) to meet the security goal level. In addition, it is easily possible to change the rule base in Prolog or improve the rules as needed (for example, if additional security features need to be evaluated or if new facts are available). That is why Prolog is a suitable tool for the initial creation of a means of manipulating the properties of objects and relations between them, which is the main subject of our study.

The purpose of the Prolog program is to support automatic evaluation of security goals. In Prolog, the logic of a program is expressed in terms of facts and rules. The program begins

with a request that the Prolog engine tries to satisfy by checking the available facts and rules. Facts and rules are the rules for determining the security goal presented earlier. For each security objective (integrity, confidentiality, and availability), a security rating (level 1–3) was assigned to EP, PP, CC facts. These facts are checked by Prolog rules to determine if the request is true. For rule set table rules representing a condition, a fact is used, for example, PP is ok (true) when the security function "authorization" evaluates to A0. When programming in Prolog, this is displayed as auth(a0). For rules that are a conditional statement, a fact with arguments is used, e.g. PP is ok (true) when the security feature "authorization" is rated A0 and "access" is rated AC1 – auth_access (a0, acl). To represent dependencies between conditions in a conditional statement, rules were used to define integrity, confidentiality, and availability. In other words, both of the above facts are combined by a logical union ("and", represented by a comma) in the rule: integ (A, Ac): auth (A), auth_access (A, Ac).

Prolog's rules and logical conjunctions allow to combine facts and conditional statements. They can include or exclude conditions or form a new condition.

The security requirements reflect the security needs of the business and determine whether a given vulnerability poses a security threat to the business. Information asset security requirements are evaluated in the context of the business process model to determine if the security functions are implemented and working correctly. The security requirements assessment considers systems, people, and the physical parts of business processes, as well as IT processes. It is shown that risk assessment techniques can benefit from an explicit assessment of the security requirements in the business context during risk identification in order to eliminate vulnerability identification errors and determine the value of the security criterion. To determine the security level of critical business processes, consider the block diagram of the concept, which is shown in **Fig. 2.9**.

The security level assessment should take into account organizational, software, technical, informational, technological and even financial issues, providing a view of the risk on a company-wide scale. These components of the functioning of the organization can be combined within the business process model. So the block diagram of a business process reflects the technological aspects of the organization's functioning, linking individual operations with "input-output" links. Such links in the proposed model correspond to connection points (CC). The business transaction itself is assigned a process point (PP). The business operation is executed based on the data received from the "control" input, which defines the normative basis of the process. The performer of a business transaction is identified by the "engine" input. Business processes as a whole are defined within the framework of the concept in the part "Formation of security contours", individual components – within the part "Assessment of the degree of implementation of security services". The task of the desired level of security for the entire system of business processes of the organization is carried out in the "Evaluation of cyber threats" part of the proposed concept. Thus, within the framework of the proposed concept, the corresponding actions related to determining the achievability of a given security level are divided into levels of the concept.

As an example, a conditional example of making a payment in online banking is considered.

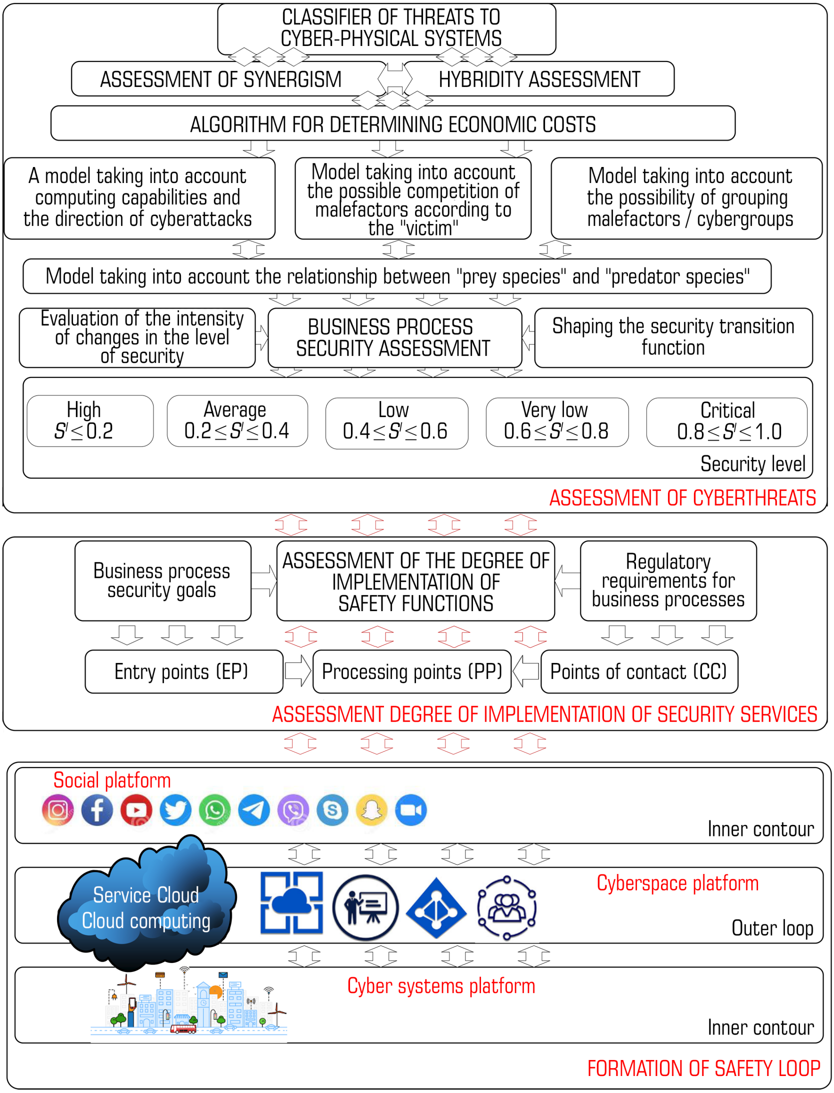


Fig. 2.9 Structural diagram of the Concept for determining the level of security to identify threats and form multi-loop security systems, taking into account the integration of technologies and the formation of hybrid-cyber-physical / socio-cyber-physical systems, it is possible to use the classifier and expert evaluation proposed in [36, 37]

The use of the proposed concept begins with the definition of critical business processes (online payment execution process). The following information assets are defined for the selected business process: customer and payment data. Customer data is stored and processed, as well as payment data necessary for transactions. The criteria and indicators for identifying these information assets are decision points and actions in the process, such as "Enter personal and payment data", "Verify personnel and payment data", or "Save personnel, contract and payment data".

The next step is to specify the information asset security requirements. First, an appropriate security goal rating for integrity, confidentiality, and availability must be selected based on the information asset's security needs. Second, the security requirements of the container must be specified; a description of what needs to be protected, as well as a specific implementation in the containers that process the information.

The company's security policy, organizational procedures, and security best practices can be used to identify and define security requirements. The security requirements defined for customer and payment data are presented in **Tables 2.5** and **2.6**.

Table 2.5 Customer data security requirements

Information asset: Customer data	Integrity	Confidentiality	Availability	IT Security Processes
Processing				
Data	I-L2	C-L2	A-L3	n/a
Containers				
Primary Systems	Address data has to be verified in the system. Data in the system should be protected against unauthorized access and modification. 192-bit AES encryption if data is transferred	Access should be given only to company people. Changes have to be logged	Within one business day	– n/a Access Management (authorizations); – IT Security Management (Security of systems); – Continuity management and Disaster Recovery; – Change Management
Organization People Process	Personnel entering data should verify their entries as well as the data received	People of the departments should be aware of confidentiality	Core people within one business day	– Access Management; – IT Security training
Physical	None	Documents should be locked away and disposed of securely	Within one business day	– IT Security training; – Facility Management; – Continuity Management

The degree of implemented security functions in the processing of information and their compliance with the security objectives of information assets is assessed. Information asset containers (systems, actors, and environment) are evaluated based on information asset security requirements. The points where information assets are created, processed or transferred (business process points EP, PP and CC) are also defined. For each EP, PP, and CC, the extent to which the security function is implemented (including access control, authorization, data validation, and communication security) is defined.

Table 2.6 Payment data security requirements

Information assets: payment data	**Integrity**	**Confidentiality**	**Availability**	**IT Security Processes**
Processing				
Data	I-L2	C-L2	A-L2	n/a
Containers				
Primary Systems	Address data has to be verified in the system. Data in the system should be protected against unauthorized access and modification. 192-bit AES encryption if data is transferred	Access should be given only to company people. Changes have to be logged	Within one business day	– Access Management (authorizations); – IT Security Management (Security of systems); – Continuity management and Disaster Recovery; – Change Management
Organization People Process	Personnel entering data should verify their entries as well as the data received	People of the departments should be aware of confidentiality	Core people within one business day	– Access Management; – IT Security training
Physical	None	Documents should be locked away and disposed of securely	Within one business day	– IT Security training; – Facility Management; – Continuity Management

Container security requirements are assessed in terms of security at each technological operation in which data is processed (points EP, PP, CC). Evidence of whether the security requirements for the banking system, organization, or cyber-physical environment are met can be obtained from the system configuration or specification, company security policy, process documentation, or other implementation examples.

2.3 CONCEPTUAL FOUNDATIONS OF THE TWO-CONTOUR CPS SECURITY SYSTEM BASED ON POST-QUANTUM ALGORITHMS

The development of wireless technologies has significantly expanded the range of digital services based on integration, cyber-physical systems with LTE (Long-Term Evolution), IEEE 802.16, IEEE 802.16e, IEEE 802.15.4, IEEE 802.11, Bluetooth technologies. Further development of this direction makes it possible to form smart-city digital services based on the synthesis of the Internet of things, mesh networks and smart technologies. The use of wireless communication channels can significantly increase the speed of information transfer and ensure the creation of socio-cyber-physical systems based on the LTE, IEEE 802.16, IEEE 802.16e, IEEE 802.15.4, IEEE 802.11, Bluetooth standards. To provide security services in cyber-physical systems, as a rule, the KNX / EIB (European Installation Bus) standard (ISO / IEC 14543) is used based on the use of virtual private network channels (encryption AES-128, -256) [38–44]. The KNX standard in IP Secure mode allows to provide confidentiality and authenticity services based on the use of an additional mechanism (protective shell) that protects all KNXnet / IP data traffic [45]. KNX Data Secure mode uses a longer KNX frame format based on CCM (Code Block Chain Message Authentication Counter) with 128-bit AES block symmetric cipher encryption to ensure information integrity. However, the security mechanisms of the KNX standard (ISO / IEC 14543) do not provide protection against IP networks channel monitoring, which allows an attacker based on a false server to intercept the traffic of the information flow. In addition, according to experts from the National Institute of Standards and Technology (NIST) of the United States, the use of symmetric algorithms with a key length of 128 bits does not provide the required level of security in the post-quantum period [46–48].

In the wireless channels of LTE technologies, only the 3A authentication service (AAA – authentication, authorization, accounting) is provided based on the Diameter protocol [49–53]. The Diameter protocol has a predefined set of common AVPs (Attribute-Value-Pair) between two Diameter nodes, allowing various combinations of the protocol to be used. However, the absence of security mechanisms in wireless channels of mobile technologies does not allow confidentiality and integrity services to be provided. As practice shows, in networks based on the Diameter protocol, attacks aimed at denial of service, disclosure of information about subscribers and the operator's network, as well as fraud against the operator are possible [54, 55]. In addition, an attacker can forcibly transfer the subscriber's device to 3G mode (third generation) – and carry out further attacks on the less secure SS7 system (signaling system, Signaling System No. 7) [55].

Table 2.7 shows the main characteristics of wireless mobile and computer networks and security services based on the KNX standard and the Diameter protocol. Analysis of the **Table 2.7** shows that in the context of the emergence of a full-scale quantum computer, hybridity and synergy of cyber-attacks, wireless channels partially provide privacy services, which requires new approaches to security based on post-quantum mechanisms. The provision of security services in wireless channels is associated with a contradiction between the speed indicators of wireless channels and the need to use cryptographic algorithms to provide confidentiality and integrity services.

Table 2.7 Wireless Network Specifications Table

Technology	Trans-mission / reception range, m	V, bps	Topology	Trans-mission spectrum	Modulation	Security services									
						before PQ					in PQ				
						C	I	A	Au	B	C	I	A	Au	B
LTE (4G)	up to 13400	up to 100 Mbps	AIPN	600 MHz before 2,5 GHz	64QAM	–	–	+	+	–	–	–	–	–/+	–
LTE (5G)	500	20 Gbps	Heterogeneous core network	from 30 GHz before 300 GHz	256-QAM	–	–	+	+	–	–	–	–	–/+	–
IEEE 802.11ac (WiFi 5)	500	up to 7 Gbps	P2MP	5 GHz	256-QAM	+	+	+	+	–	–	–	–	–/+	–
IEEE 802.11ax (Wi-Fi 6)	–	9607 Mbps	P2MP	5 GHz	1024-QAM	+	+	+	+	–	–	–	–	–/+	–
IEEE 802.16	5000	32 Mbps 134 Mbps	mesh	10-66 GHz	64QAM O-QPSK	+	+	+	+	–	–	–	–	–/+	–
IEEE 802.16m (WiMAX2)	6000	90 Mbps 179 Mbps	mesh	11 GHz	64QAM	+	+	+	+	–	–	–	–	–/+	–
IEEE 802.15.1 (Bluetooth 5)	200	2-6 Mbps	mesh	2.4-2.485 GHz	64QAM	+	+	+	+	–	–	–	–	–/+	–
IEEE 802.15.4	1000	250 kbps	P2P Cluster tree	2.4-2.483 GHz	BPSK O-QPSK	+	+	+	+	–	–	–	–	–/+	–

Note: C – confidentiality; I – integrity; A – availability; Au – authenticity; B – involvement; AIPN – All IP Network; P2MP –point-to-multipoint; P2P – peer-to-peer; QAM – Quadrature Amplitude Modulation; O-QPSK – quadrature phase shift keying; BPSK– binary phase shift keying

In addition, in the post-quantum cryptoperiod, the requirements for symmetric cryptography algorithms increase significantly, which casts doubt on the possibility of providing a compromise between the amount of key data and the amount of memory of switching devices. Also, the possibility of offline encryption of various information flows based on symmetric cryptography. To provide security services, it is proposed to use post-quantum algorithms – crypto-code constructions of McEliece and Niederreiter on algebrogeometric codes [56–60]. Both crypto-code constructions are based on the principle of using the theory of error-correcting coding and the orthogonality of the matrices G, the generating matrix of the linear code, and H, the check matrix of the linear code. Taking into account the orthogonality of matrices G and H ($G \times H^T = 0$), these cryptosystems have almost the same energy costs for encryption (**Fig. 2.10**).

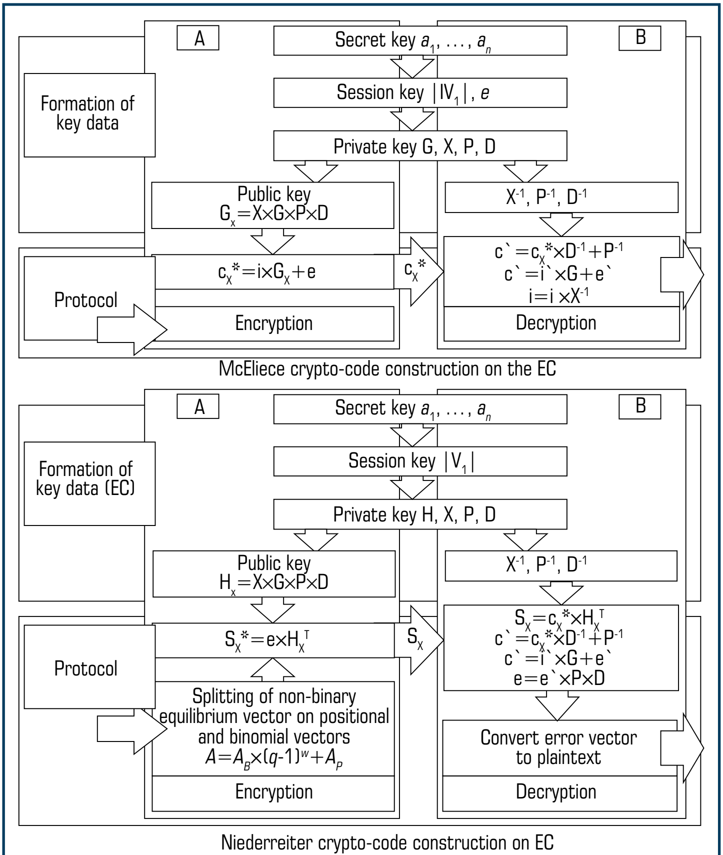


Fig. 2.10 Block diagrams of McEliece and Niederreiter CCC

Concealment matrices are used as a key sequence in both crypto-code constructions:

– *X* – masking nondegenerate randomly equiprobable $k \times k$ matrix formed by the source of keys with elements from $GF(q)$;

– *P* – permutation randomly equiprobably formed by the source of keys $n \times n$ matrix with elements from $GF(q)$;

– *D* – diagonal matrix formed by the $n \times n$ key source with elements from $GF(q)$;

– *G* – $k \times n$ generating matrix (McEliece CCC);

– *H* – check matrix with dimension $r \times n$. In addition, a distinctive feature of Niederreiter's *CCC* is the preliminary use of equilibrium coding, which makes it possible to provide a practically relative coding rate equal to one.

Table 2.8 shows the main characteristics of elliptic (EC) modified elliptic (MEC) codes. Notations: $GF(q)$ – *Galois* finite field; *X* – smooth projective algebraic curve in a projective space P^n over $GF(q)$, $g = g(X)$ – kind of curve; $X(GF(q))$ – the set of its points over a finite field; $N = X(GF(q))$ – their number. C – divisor class on *X* power $\alpha > g-1$, *C* defines the mapping $\varphi: X \rightarrow P^{k-1}$, where $k \geq \alpha - g + 1$. The set $y_i = \varphi(x_i)$ specifies the code. Number of points in intersection $\varphi(X)$ with the hyperplane is equal to α, i.e. $n - d \leq \alpha$. This construction allows building codes with parameters $k + d \geq n - g + 1$, length *n* of which is less than or equal to the number of points on the curve *X*.

Table 2.8 EC, MEC main characteristics (*n, k, d*)

Characteristics	**EC**	
(*n*, *k*, *d*) parameters of the code that is built through of the view $\varphi: X \rightarrow P^{k-1}$	$n = 2\sqrt{q} + q + 1$, $k \geq \alpha$, $d \geq n - \alpha$, $\alpha = 3 \times degF$, $k + d \geq n$	
(*n*, *k*, *d*) parameters of the code that is built through of the view $\varphi: X \rightarrow P^{r-1}$	$n = 2\sqrt{q} + q + 1$, $k \geq n - \alpha$, $d \geq \alpha$, $\alpha = 3 \times degF$, $k + d \geq n$	
Characteristics	**Shortened MEC**	**Extended MEC**
(*n*, *k*, *d*) parameters of the code that is built through mapping of the view $\varphi: X \rightarrow P^{k-1}$	$n = 2\sqrt{q} + q + 1 - x$, $k \geq \alpha - x$, $d \geq n - \alpha$, $\alpha = 3 \times degF$, $k + d \geq n$	$n = 2\sqrt{q} + q + 1 - x + x_1$, $k \geq \alpha - x + x_1$, $d \geq n - \alpha$, $\alpha = 3 \times degF$
(*n*, *k*, *d*) parameters of the code that is built through mapping of the view $\varphi: X \rightarrow P^{r-1}$	$n = 2\sqrt{q} + q + 1 - x$, $k \geq n - \alpha$, $d \geq \alpha$, $\alpha = 3 \times degF$, $k + d \geq n$	$n = 2\sqrt{q} + q + 1 - x + x_1$, $k \geq n - \alpha$, $d \geq \alpha$, $\alpha = 3 \times degF$

Table 2.9 shows the main parameters of the McEliece cryptosystem. The paper [21] presents the results of studies of the energy consumption of crypto transformations and cryptographic strength. The obtained results confirm the possibility of practical implementation and use of post-quantum algorithms (McEliece and Niederreiter CCC) in wireless channels in offline mode.

Table 2.9 The main parameters of McEliece CCC on EC, MEC

Parameter	CCC EC	
private key size	$l_{K+}=n^2\times k^2\times m$	
information vector size	$l_I=k\times m$	
cryptogram dimension	$l_s=n\times m$	
relative transfer rate	$R=l_I/l_s={}^{k\times m}\!/_{n\times m}$	
Parameter	**CCC on shortened MEC**	**CCC on extended MEC**
private key size	$l_{K+}=x\times\left\lceil\log_2\left(2\sqrt{q}+q+1\right)\right\rceil$	$l_{K+}=\left(x-x_1\right)\times\log_2\left(2\sqrt{q}+q+1\right)$
information vector size	$l_I=(\alpha-x)\times m$	$l_I=(\alpha-x+x_1)\times m$
cryptogram dimension	$l_S=\left(2\sqrt{q}+q+1-x\right)\times m$	$l_S=\left(2\sqrt{q}+q+1-x+x_1\right)\times m$
relative transfer rate	$R=\left(\alpha-x\right)/\left(2\sqrt{q}+q+1-x\right)$	$R=\left(\alpha-x+x_1\right)/\left(2\sqrt{q}+q+1-x+x_1\right)$

When developing the conceptual foundations of the two-contour CPS security system based on post-quantum algorithms, the approach proposed in [21] was used. It is based on the concept of double-contour security based on crypto-code constructions. At the same time, it is proposed to use integrated solutions for the use of certain codes in crypto-code systems based on the gradation of the information secrecy degree in socio-cyber-physical systems. **Table 2.10** shows the ratio of time and the degree of information secrecy.

Table 2.10 The ratio of time and the degree of information secrecy

The degree of information secrecy	Time	Suggested codes for CCC
critical	up to 1 year	MEC, flawed codes
high	up to 1 month	MEC
medium	up to 1 hour	EC
low	up to 10 minutes	EC
very low	up to 1 minute	LDPC

Fig. 2.11 shows a block diagram of the conceptual foundations of a dual-contour security system. For a formal description of the Concept, the approach in [21], will be used: to ensure the

security of the entire protection system, it is necessary to take into account the threats of the internal and external contours for each of the platforms:

1) threats of the internal contour, taking into account the hybridity and synergy of threats for the CPS platform:

$$W^{!\,PS\ ISL}_{\text{hybrid}\,!,I,Au\,synerg_{\text{CPS platform}}} = W^{!\,PS\ ISL}_{synerg_{\text{CPS platform}}}{}^{!} \cap W^{CPS\ ISL}_{synerg_{\text{CPS platform}}}{}^{I} \cap W^{CPS\ ISL}_{synerg_{\text{CPS platform}}}{}^{Au}, \tag{2.1}$$

where $W^{!\,PS\ ISL}_{synerg_{\text{CPS platform}}}{}^{!}$ – synergy of threats on privacy service, $W^{!\,PS\ ISL}_{synerg_{\text{CPS platform}}}{}^{I}$ – synergy of threats on integrity service, $W^{!\,PS\ ISL}_{synerg_{\text{CPS platform}}}{}^{Au}$ – synergy of threats on authenticity service;

2) threats of the internal contour, taking into account the hybridity and synergy of threats for the cyberspace platform (CbS):

$$W^{!\,bS\ ISL}_{\text{hybrid}\,!,I,Au\,synerg_{\text{CbS platform}}} = W^{!\,bS\ ISL}_{synerg_{\text{CbS platform}}}{}^{!} \cap W^{CbS\ ISL}_{synerg_{\text{CbS platform}}}{}^{I} \cap W^{CbS\ ISL}_{synerg_{\text{CbS platform}}}{}^{Au}, \tag{2.2}$$

where $W^{!\,bS\ ISL}_{synerg_{\text{CbS platform}}}{}^{!}$ – synergy of threats on privacy service, $W^{!\,bS\ ISL}_{synerg_{\text{CbS platform}}}{}^{I}$ – synergy of threats on integrity service, $W^{!\,bS\ ISL}_{synerg_{\text{CbS platform}}}{}^{Au}$ – synergy of threats on authenticity service.

Overall threat assessment of the inner loop, taking into account CPS technologies:

$$W^{CPS}_{ISL} = W^{!\,PS\ ISL}_{\text{hybrid}\,!,I,Au\ synerg_{\text{CPSplatform}}} \cup W^{!\,bS\ ISL}_{\text{hybrid}\,!,I,Au\ synerg_{\text{CbSplatform}}}. \tag{2.3}$$

General assessment of threats of the internal contour, taking into account the form of ownership of the elements and technologies of the cyber-physical system (**Fig. 2.11**):

$$W^{CPS}_{ISL_{\text{general}}} = W^{CPS}_{ISL_{\text{private.}}} \cup W^{CPS}_{ISL_{\text{state}}} \cup W^{CPS}_{ISL_{\text{corporativ}}}, \tag{2.4}$$

where $W^{CPS}_{ISL_{\text{private.}}}$ – overall assessment of internal contour threats to the personal property system, $W^{CPS}_{ISL_{\text{state}}}$ – overall assessment of threats of the internal contour for the state property system, $W^{CPS}_{ISL_{\text{corporativ}}}$ – overall assessment of internal contour threats to the corporate property system;

3) threats of the external contour, taking into account the hybridity and synergy of threats for the CPS platform:

$$W^{!\,PS\ \ SL}_{\text{hybrid}\,!,I,Au\,synerg_{\text{CPS platform}}} = W^{!\,PS\ \ SL}_{synerg_{\text{CPS platform}}}{}^{!} \cap W^{CPS\ \ SL}_{synerg_{\text{CPS platform}}}{}^{I} \cap W^{CPS\ \ SL}_{synerg_{\text{CPS platform}}}{}^{Au}, \tag{2.5}$$

where $W^{!\,PS\ \ SL}_{synerg_{\text{CPS platform}}}{}^{!}$ – synergy of threats on privacy service, $W^{!\,PS\ \ SL}_{synerg_{\text{CPS platform}}}{}^{I}$ – synergy of threats on integrity service, $W^{!\,PS\ \ SL}_{synerg_{\text{CPS platform}}}{}^{Au}$ – synergy of threats on authenticity service;

4) threats of the external contour, taking into account the hybridity and synergy of threats for the cyberspace platform (CbS):

$$W^{!\,bS\ ESL}_{\text{hybrid}\,!,I,Au\,synerg_{\text{CbS platform}}} = W^{!\,bS\ ESL}_{synerg_{\text{CbS platform}}}{}^{!} \cap W^{CbS\ ESL}_{synerg_{\text{CbS platform}}}{}^{I} \cap W^{CbS\ ESL}_{synerg_{\text{CbS platform}}}{}^{Au}, \tag{2.6}$$

where $W^{!\,bS\,ESL\quad !}_{synerg_{CbS\ platform}}$ – synergy of threats on privacy service, $W^{!\,bS\,ESL\quad I}_{synerg_{CbS\ platform}}$ – synergy of threats on integrity service, $W^{!\,bS\,ESL\quad Au}_{synerg_{CbS\ platform}}$ – synergy of threats on authenticity service.

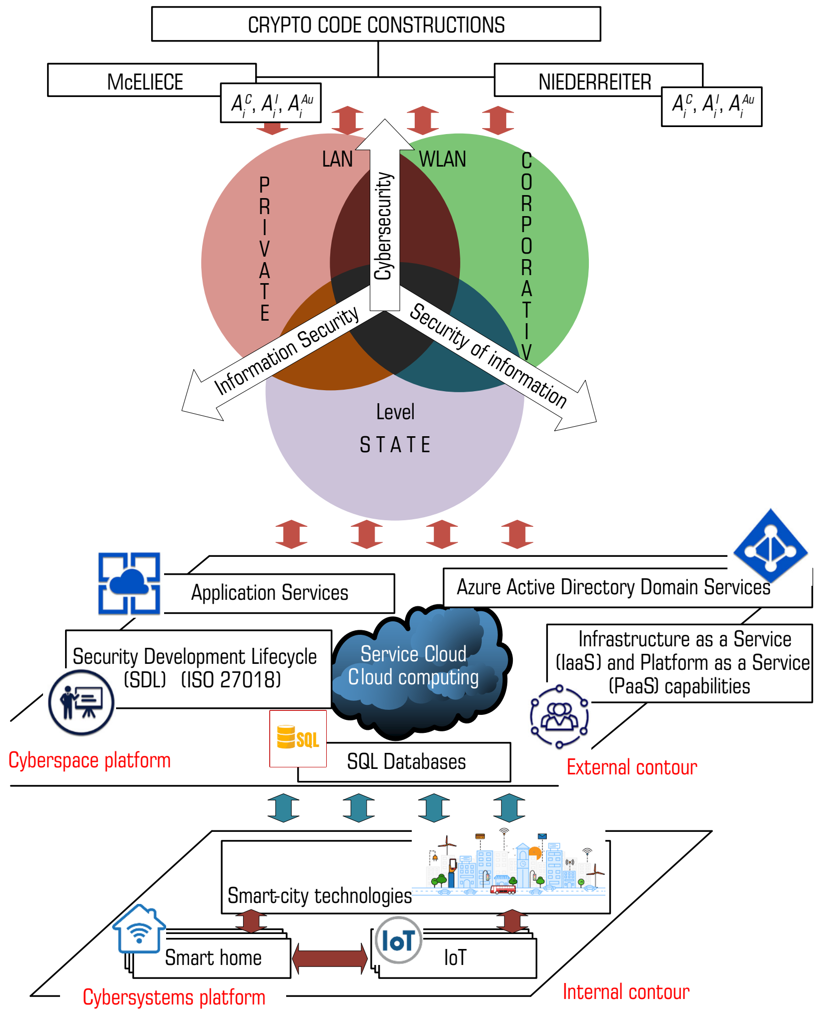


Fig. 2.11 Structural diagram of the conceptual foundations of a double-contour security system

General assessment of threats of the external contour, taking into account CPS technologies:

$$W_{SL}^{CPS} = W_{\text{hybrid}\,!\,,I,Au\ synerg_{\text{CPSplatform}}}^{!\ PS\ ESL} \cup W_{\text{hybrid}\,!\,,I,Au\ synerg_{\text{CbSplatform}}}^{!\ bS\ ESL}. \quad (2.7)$$

Overall assessment of threats of the internal contour, taking into account the form of ownership of the elements and technologies of the cyber-physical system (**Fig. 2.11**):

$$W_{SL_{\text{general}}}^{CPS} = W_{SL_{\text{private.}}}^{CPS} \cup W_{SL_{\text{state}}}^{CPS} \cup W_{SL_{\text{corporativ}}}^{CPS}, \quad (2.8)$$

where $W_{SL_{\text{private.}}}^{CPS}$ – overall assessment of internal contour threats to the personal property system, $W_{SL_{\text{state}}}^{CPS}$ – overall assessment of threats of the internal contour for the state property system, $W_{SL_{\text{corporativ}}}^{CPS}$ – overall assessment of internal contour threats to the corporate property system.

Based on expressions (2.3), (2.7), an assessment of threats in cyber-physical systems in the internal and external CPS security contours is formed, and on the basis of expressions (2.4), (2.8) – taking into account forms of ownership (separately). To provide a generalized assessment of a multicontour security system, the formula was used:

$$W_{\text{final}}^{CPS} = W_{ISL_{\text{general}}}^{CPS} \cup W_{ESL_{\text{general}}}^{CPS}. \quad (2.9)$$

Required information resource security services $I_{A_i} \in \{I_A\}$ can be described by a tuple $I_{A_i} = \left(Type_i, A_i^C, A_i^I, A_i^{Au}, \beta_i\right)$.

$Type_i$ – information asset type, described by a set of basic values: $Type_i = \{CI_i, PD_i, CD_i, TS_i, StR_i, PubI_i, ContI_i, PI_i\}$, where CI_i – confidential information, PD_i – payment documents, CD_i – loan documents, TS_i – commercial secret, StR_i – statistical reports, $PubI_i$ – public information, $ContI_i$ – control information, PI_i – personal data, β_i – a metric of the ratio of time and degree of information confidentiality for an asset (critical – 1.0; high – 0.75; medium – 0.5; low – 0.25; very low – 0.01).

Then the general (current) level of security of cyber-physical systems based on wireless mobile technologies is described by the expression:

– for additive convolution:

$$L_{W_{\text{security}}^{CPS}} = L_{ISL}\sum_{j=1}^{2}\sum_{i=1}^{8}\left(I_{A_{ij}} \times \beta_{ij}\right) + L_{ESL}\sum_{j=1}^{2}\sum_{i=1}^{8}\left(I_{A_{ij}} \times \beta_{ij}\right). \quad (2.10)$$

– for multiplicative convolution:

$$L_{W_{\text{security}}^{CPSS}} = 1 - \left[1 - L_{ISL}\sum_{j=1}^{2}\sum_{i=1}^{8}\left(I_{A_{ij}} \times \beta_{ij}\right)\right] \times \left[1 - L_{ESL}\sum_{j=1}^{2}\sum_{i=1}^{8}\left(I_{A_{ij}} \times \beta_{ij}\right)\right]. \quad (2.11)$$

Such an approach will allow to provide the required level of security in a timely manner, taking into account the degree of information secrecy and / or the security time that is necessary to provide confidentiality, integrity and authenticity services. To ensure the required security level, it is possible to combine various codes and CCC, both McEliece and Niederreiter.

2.4 MATHEMATICAL MODEL OF A METHOD FOR ENSURING CONFIDENTIALITY AND AUTHENTICITY IN WIRELESS CHANNELS

The formation of a method for ensuring the confidentiality and authenticity of information based on a two-contour security system is proposed to be implemented at the CCC. The use of various error-correcting codes, defective and LDPC codes in McEliece and Niederreiter CCC allows to form various combinations, taking into account the level of secrecy (confidentiality), as well as the required time of information cryptographic strength. **Fig. 2.12** shows a block diagram of the proposed approach.

The main elements of the internal contour are various physical control devices (sensors, counters, tracking sensors, video cameras, etc.), as well as the server for dispatching and controlling the physical mechanisms of the CPS. The elements of the outer contour include a server for generating key sequences and storing long-term keys, as well as mobile applications (if necessary) for CPS users.

To provide security services, it is proposed to install McEliece CCC software and hardware encoders on the elements of the internal contour. In this case, on the basis of complexing and the level of circulating information secrecy, various noise-immune codes are established and used. The dependence of time and the degree of information secrecy is given in **Table 2.9**.

The relationship between the internal and external contours of the security system on the basis of the proposed method is provided by Niederreiter's CCC, the relative coding rate is close to 1. At the same time, the use of two CCCs increases the security level by at least 2 times, and the use of various error-correcting codes integrated provides an increase in the error reliability.

Long-term keys are formed in the external contour, the use of which allows to reset the CPS security system (in case of loss of gadgets control, compromise, etc.). As well as the formation of private keys for use in the CPS in the internal contour. To ensure security, long-term keys are stored in encrypted form by McEliece or Niederreiter CCC. This approach ensures the closure of all channels of information transmission both in the internal contour – the cyber-physical system, and in the external contour, as well as communication channels between the contours. In this case, post-quantum cryptosystems are used, which makes it possible to use this approach with the prospect of a full-scale quantum computer.

To form a mathematical model of the method for ensuring confidentiality and authenticity in wireless channels, the approaches of the works [21, 56–60] will be used.

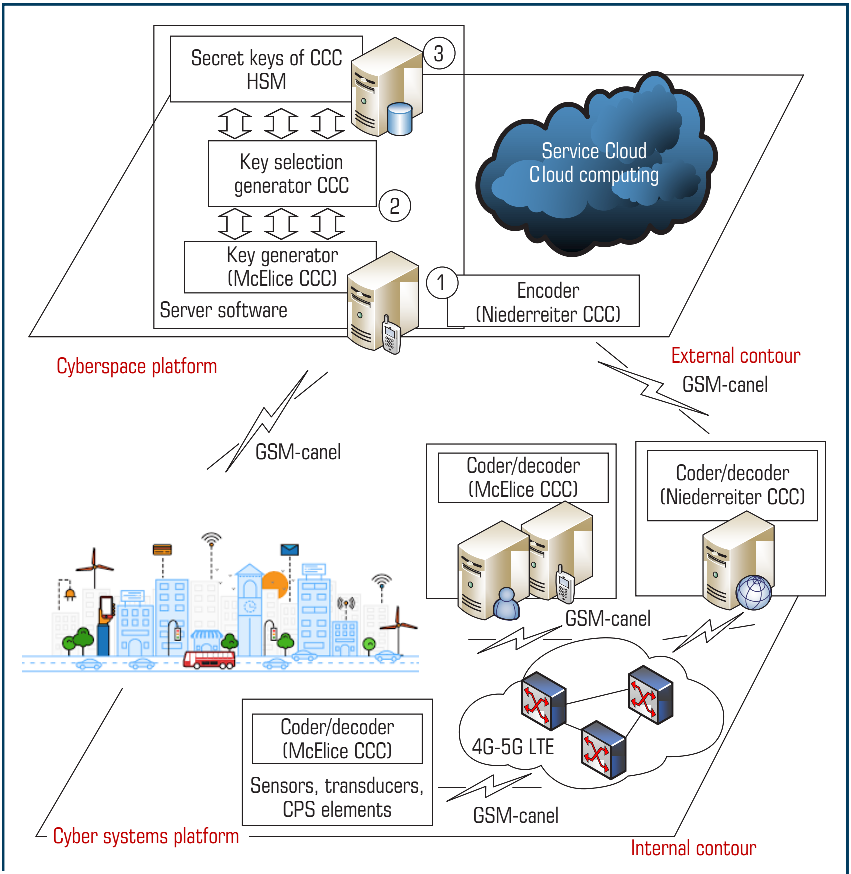


Fig. 2.12 Structural diagram of the formation of a method for ensuring confidentiality and data integrity

The initial data for the mathematical models of McEliece and Niederreiter CCC are:

– set of open texts for McEliece CCC $M=\{M1, M2, \dots, M_{q^k}\}$, where $M_i=\{I_0, I_{h_1},\dots, I_{h_j}, I_{k-1}\}$, h_j – information characters equal to zero, $|h|=1/2k$, i.e. $I_i=0$, $\forall I_i \in h$; for Niederreiter CCC $_i=\left\{e_0, e_{h_1,}\dots e_{h_k}, e_{e-1}\right\}, \forall e_e \in GF(q)$, h_e – error vector characters that are equal to zero, $|h|=1/2e$, that is $e_i=0$, $\forall e_i \in h$. Based on the equilibrium coding algorithm, the plaintext is converted into an error vector;

– set of closed texts (codegrams) for McEliece CCC $C=\{C_1, C_2, \dots, C_{q^k}\}$, where $C_i=(A^*_{X_0}, A^*_{h_1}, \dots, A^*_{h_j}, A^*_{X_{g-1}})$, $\forall A^*_{X_j} \in GF(q)$; for Niederreiter CCC $S=\left\{S_0, S_1, \dots S_{q^r}\right\}$, where $S_i=\left\{S^*_{X_0}, S^*_{h_1}, \dots S^*_{h_j}, S^*_{X_r}\right\}$, $\forall S_{X_r} \in GF(q)$;

– a set of direct mappings (based on the use of a public key – generating / checking matrix of error-correcting codes (error-correcting code – ECC) (algebrogeometric codes: EC, MEC; LDPC; flawed codes):

1) for McEliece CCC – $\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_s)$, where $\phi_i : M \to C_{k-h_j}$, $i = 1, 2, \ldots, s$;

2) for Niederreiter CCC – $\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_r)$, where $\varphi_i : M \to S_{r-h_e}$, $i = 1, 2, \ldots, e$;

– set of inverse mappings (based on the use of a private key – masking matrices):

1) for McEliece CCC – $\phi^{-1} = \{\phi_1^{-1}, \phi_2^{-1}, \ldots, \phi_s^{-1}\}$, where $\phi_i^{-1} : C_{k-h_j} \to M$, $i = 1, 2, \ldots, s$;

2) for Niederreiter CCC – $\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_r)$, where $\varphi_i : M \to S_{r-h_e}$, $i = 1, 2, \ldots, e$;

– set of keys parameterizing direct mappings (authorized user's public key):

1) for McEliece CCC – $KU_i = \{KU_1, KU_2, \ldots, KU_S\} = \{G_1^{ECC}, G_2^{ECC}, \ldots, G_S^{ECC}\}$, where $G_{X_{a_i}}^{LDPC_i}$ – generating $n \times k$ matrix disguised as a random code. The matrix is determined from the orthogonality of the generator and check matrices;

2) for Niederreiter CCC – $KU_i = \{KU_1, KU_2, \ldots, KU_r\} = \{H_1, H_2, \ldots, H_r\}$, where $H_{X_{a_i}}^{ECC_i}$ – erification $(N–K)\times N$ matrix determines $(N–K)$ checking symbols $P_1, P_2, \ldots, P_{N–K}$ as a linear combination of information symbols d_k, $k = 1, 2, \ldots, K$;

– set of personal (private) keys of users:

$$KR = \{KR_1, KR_2, \ldots, KR_r\} = \left\{ \begin{matrix} \{X, P, D\}_1, \\ \{X, P, D\}_2, \ldots, \{X, P, D\}_r \end{matrix} \right\}, \{X, P, D\}_i = \{X^i, P^i, D^i\},$$

where X^i – masking non-degenerate randomly equally probable generated by the key source $k\times k$ matrix with elements from $GF(q)$; P^i – permutable randomly equiprobably formed by the source of the keys $n\times n$ matrix with elements from $GF(q)$; D^i – diagonal formed by the source of the keys $n\times n$ matrix with elements from $GF(q)$. Due to the fact that the diagonal matrix is equal to the identity matrix, the value can be neglected, which reduces the capacity and complexity of the calculation.

The public key is formed by multiplying the masking matrices by the generator/checking matrices:

– for McEliece CCC – $G_{X_{a_i}}^{ECCu} = X^u \times G^{ECCu} \times P^u, u \in \{1, 2, \ldots, s\}$;

– for Niederreiter – $H_{X_{a_i}}^{ECCu} = X^u \times H^{ECCu} \times P^u, u \in \{1, 2, \ldots, r\}$.

The communication channel receives:

– for McEliece CCC – codeword: <<Eqn0072.eps>> where e – additional session key of each information package.

– for Niederreiter CCC – syndromic sequence:

$$S^* = (e_n) \times H_{X_{a_i}}^{ECC^T}.$$

On the receiving side, an authorized user who knows the concealment matrices uses a fast decoding algorithm:

– for McEliece CCC:

$$M_i = f_u^{-1}\left(C_j^*, \{X, P, D\}_u\right).$$

To recover the plaintext, the authorized user adds null information characters $C_j^* = C_j + C_{k-h_j}$, from the restored private text C_j removes the action of the secret permutation and diagonal matrices P^u and D^u.

$$C = C_j^* \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} = \left(M_i \times \left(G_{X_{a_i}}^{ECCu}\right)^T + e \right) \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} =$$

$$= \left(M_i \times \left(X^u \times G_{X_{a_i}}^{ECCu} \times P^u \times D^u\right)^T + e \right) \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} =$$

$$= M_i \times \left(X^u\right)^T \times \left(G_{X_{a_i}}^{ECCu}\right)^T \times \left(P^u\right)^T \times \left(D^u\right)^T \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} + e \times \left(D^u\right)^{-1} \cdot \left(P^u\right)^{-1} =$$

$$= M_i \times \left(X^u\right)^T \times \left(G_{X_{a_i}}^{ECCu}\right)^T + e \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1},$$

decodes the resulting vector using the Berlekamp-Massey algorithm [32]:

$$! = M_i \times \left(X^u\right)^T \times \left(G_{X_{a_i}}^{ECCu}\right)^T + e \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1},$$

i.e., gets rid of the second term and the factor $\left(G_{X_{a_i}}^{ECCu}\right)^T$ in the first term on the right side of the equality, after which it removes the effect of the masking matrix X^u. To do this, the result of decoding $M_i \times (X^u)^T$ should be multiplied by $(X^u)^{-1}$: $(M_i \times (X^u))^T \times (X^u)^{-}1 = M_i$. The resulting solution is the essence of plain text M_i;

– for Niederreiter CCC.

Next, an authorized user, using a set of matrices $\{X, P, D\}_u = \{X^u, P^u, D^u\}$ forms a vextor: $\bar{A}^* = A_X^* \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1}$, thus unmasks the code sequence $A_{X_i}^*$.

After substitution, getting the equality:

$$\bar{A}^* = A_X^* \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} = \left(A_{X_i} + M_i\right) \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} =$$

$$= A_{X_i} \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} + M_i \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1}.$$

An authorized user who has generated a vector has the ability to apply a fast (polynomial complexity) noise-correcting decoding algorithm and thus generate a vector $\bar{A}^* = A_X^* \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1}$ and vector $M_i^u = M_i \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1}$.

To restore the information equilibrium sequence M_i it is enough again to multiply vector M_i^u by masking matrices P^u and D^u, in reverse sequence:

$$M_i = M_i^u \times P^u \times D^u = M_i \times \left(D^u\right)^{-1} \times \left(P^u\right)^{-1} \times P^u \times D^u = M_i.$$

Thus, the presented mathematical model makes it possible to use McEliece and Niederreiter CCC to provide confidentiality, integrity, and authenticity services and to practically implement the proposed method. **Fig. 2.13** presents the main elements of the proposed method for providing basic security services in cyber-physical systems based on wireless communication channels. The main difference from the known approaches is, while maintaining the level of bandwidth of the wireless channel, to provide the required level of security (cryptostrength) of the channel in an integrated way (cryptosecurity based on post-quantum algorithms at the level of 10^{25}–10^{35} group operations), (P_{err} not less than 10^{-9}–10^{-12}) [21, 56–60].

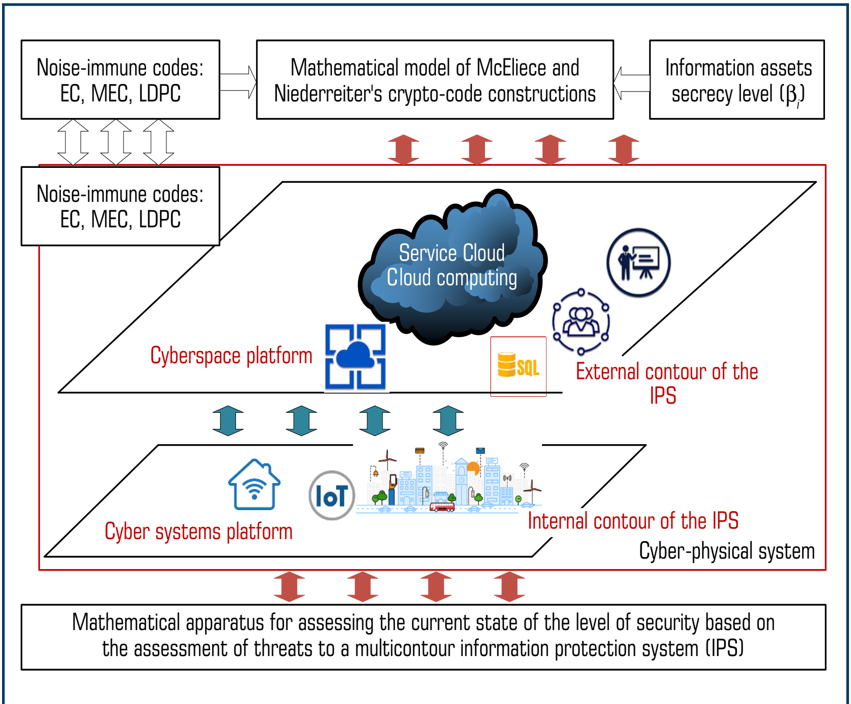


Fig. 2.13 Block diagram of the proposed method for providing security services in wireless channels based on crypto-code constructions

The use of post-quantum algorithms – crypto-code structures on the MEC allows the formation of cryptosystems over the field GF (2^6), and when using defective codes, the formation of hybrid cryptosystems over the field GF (2^4), which allows their practical implementation on resource-limited chipsets. The conducted experimental studies have shown that the use of CCC in cyber-physical systems based on wireless channels requires the presence of a mobile Internet channel (broadband channel). This limitation is based on the existing multi-contour CPS model, where, as a rule, the control system is deployed in cloud technologies.

Thus, the proposed model for the use of McEliece and Niederreiter CCC is the basic component of the proposed method for the formation of cryptosystems to provide basic security services. Such an approach, taking into account the multi-contour CPS and an objective assessment of the current state of security, makes it possible to form a reliable information protection system in the post-quantum period, taking into account possible APT attacks with signs of hybridity and synergy.

Algorithm for the practical implementation of confidentiality and authenticity in wireless channels

One of the variants of the proposed method for ensuring confidentiality and authenticity in outbred channels is the use of command transmission security based on two-way communication channels. For example, the connection between the key fob and the car's on-board computer. To form a "dialogue coding" that requires a two-way communication channel (the presence of a receiver and a transmitter, both in the main module and in the key fob), let's use McEliece CCC. In this case, the internal security contour is formed on the basis of encryption of the key fob authentication information package and information packages for the execution of various commands (unlocking the car, opening the driver's door, opening the trunk, etc.). The external contour (cyberspace platform) stores a long-term secret key that allows to reset the key sequences of the CCC and, at the request of the user, generates private keys and public keys of the CCC.

To ensure the service of authenticity – the authenticity of sending a command from the key fob of an authorized user, it is proposed to use a random number that is generated at each "appearance" by the on-board computer of the car, it generates a random number (session key) with a length of 76 bits.

As a pseudo-random number generator of length 76, it is proposed to use a pseudo-random number generator based on a linear recurrent feedback shift register (LFSR) modulo an irreducible polynomial of 76 degrees. These pseudo-random number generators generate sequences of the maximum period and are easily implemented both in hardware and software [61]. The general structural diagram of the LFSR is shown in **Fig. 2.14**.

LFSR during the first k time counts, the key (switch) is in the upper position, and the shift register is filled with the key sequence $K_i = (K_{i0}, K_{i1}, \ldots, K_{ik-1})$. During the following q^{k-1} time readings, the key (switch) is in the lower position and the values stored in the cells of the shift register are fed to the output of the device. At each time interval, it is in the lower position and the values stored in the cells of the shift register are fed to the output of the device. At each time interval, the information stored in the shift register moves one cell to the right, and the value stored in the

rightmost cell is fed through the LRFSR feedback loop. The feedback function specifies a specific type of feedback circuit switching and ensures the formation of a pseudo-random sequence of the maximum period.

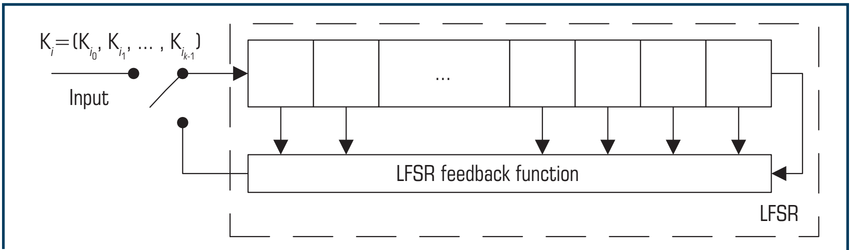


Fig. 2.14 General block diagram of a recurrent shift register with feedback

The resulting number of 76 bits is converted into an information sequence $I_{1\times19}$ with elements from the field GF (2^4), due to the use of multiplexers according to the scheme presented in **Fig. 2.15**.

Let's consider the algorithm for ensuring the authenticity and confidentiality of the formation of "dialogue coding" based on McEliece CCC. Car key fob software:

1. Sends a request to execute a command.
2. Receive over an open channel $I_{2(1\times19)}$.
3. Removing masking matrices and receiving $I_{1\times19}$:

$$I_{1(1\times19)} = I_{2(1\times19)} \times {}^{-1}_{1(1\times19)} \times D^{-1}_{1(1\times19)}.$$

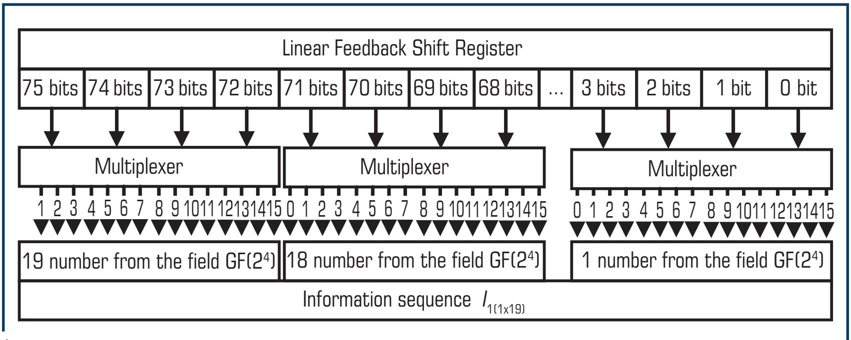


Fig. 2.15 Scheme for converting a random number into an information sequence $I_{1\times19}$ with elements from the field GF (2^4)

4. Generation of an additional session key – *e* (error vector) – corresponding to some action) (number from 0 to 24). To form the error location, it is proposed to transform the sequence $I_{1\times19}$ into a binary sequence and take the corresponding number in decimal number modulo 25.

5. Formation of a cryptogram. To do this, it is proposed to use the CCC on *EC* (*MEC*) (25, 19, 3)-code:

$$c_{1(1\times19)} = I_{1(1\times19)} \times G^{ECC}_{1(1\times19)} + e,$$

where $G^{ECC}_{1(1\times19)} = X_{1(1\times19)} \times G_{1(1\times19)} \times P_{1(1\times19)} \times D_{1(1\times19)}$, masking matrices $X_{1(1\times19)}$, $P_{1(1\times19)}$ and $D_{1(1\times19)}$ are long-term keys of the key fob and on-board computer of the car.

6. $c_{1(1\times19)}$ enters the communication channel and is transmitted to the on-board computer of the car.

Vehicle on-board computer software:

1. At the request of the car key software, a random number of 76 bits is generated.

2. Converted to information sequence I_{1x19} with elements from the field $GF\ (2^4)$.

3. Transformation of the information sequence I_{1x19} based on the use of McEliece CCC masking matrices $P_{1(1x19)}$ and diagonal matrix $D_{1(1x19)}$:

$$I_{2(1\times19)} = I_{1(1\times19)} \times {}_{1(1\times19)} \times D_{1(1\times19)}.$$

4. Transmission $I_{2(1\times19)}$ on a key fob.

5. Upon receipt $c_{1(1\times19)}$ based on the fast Berlekamp-Massey decoding algorithm, the error vector is found, which determines the command that came from the key fob.

Table 2.11 presents the results of the analysis of the provision of security services: confidentiality, integrity and authenticity using various wireless channels.

Analysis of the **Table 2.11** shows that the use of symmetric cryptosystems based on block and stream ciphers (used in the KNX standard) do not provide full confidentiality and integrity services in the post-quantum period.

Thus, the proposed algorithm ensures the closing of the wireless channel using a software and hardware complex. The use of a hardware solution for closing (encrypting) the execution command on the on-board computer of the car will counteract almost all threats of intercepting the code execution command and hacking the car's security system as a whole.

This approach confirms that the use of CCC makes it possible to provide basic security services in wireless channels without forming VPN channels, which greatly simplifies practical use in the formation of the architecture of smart-city networks. Thus, the proposed method on CCC with different ECCs makes it possible to provide not only security services, but also to increase the reliability of transmitted information flows in the channels of LTE technologies in an integrated manner. The use of a multi-loop information security system based on CCC also forms an objective assessment of threats, and the current state of security of the system as a whole. The main limitation

of the proposed approach for providing security services in wireless communication channels is the use of either a tunnel mode for point-to-point connection, or the use of a mobile broadband Internet channel, in which there are "no" possible filters and limitations of various mobile communication providers.

A promising direction for further research is to evaluate the effectiveness of using the proposed method and the Concept of multicontour information protection based on the use of McEliece and Niederreiter CCC with the formation of a control system based on a desktop server.

Table 2.11 Comparative characteristics of wireless channels

Technology	Providing security services			Degree of information secrecy (βi)				
	A_i^C	A_i^I	A_i^{Au}	**1.0**	**0.75**	**0.5**	**0.25**	**0.01**
LTE (4G), LTE (5G)	–	–	–/+	–	–	–	–	–
IEEE 802.11 ac (WiFi 5)	–	–	–/+	–	–	–	–	–
IEEE 802.11ax, Wi-Fi 6+KNX	–/+	–/+	–/+	–	–	–	+	+
IEEE 802.16+KNX	–/+	–/+	–/+	–	–	–	+	+
IEEE802.16 m (WiMAX2)	–/+	–/+	–/+	–	–	–	+	+
IEEE 802.15.1, Bluetooth 5+KNX	–/+	–/+	–/+	–	–	–	+	+
IEEE 802.15.4+KNX	–/+	–/+	–/+	–	–	–	+	+
LTE+CCC на ECC (EC)	+	+	+	+	+	+	+	+
LTE+CCC на ECC (MEC)	+	+	+	+	+	+	+	+

2.5 METHODS FOR THE PRACTICAL IMPLEMENTATION OF MCELIECE AND NIEDERREITER CRYPTO-CODE STRUCTURES

The creation of modern synthesized networks is based on the hybridization of technologies of wireless mobile and socio-cyber-physical systems based on the Internet of things. Classical computer systems and technologies integrate elements of the Internet of things and form fundamentally new directions for the development of the IT industry, smart technologies that combine all the achievements of mobile, wireless and socio-cyber-physical systems. However, the rapid expansion of mesh-, sensor networks using wireless channel standards: mobile technologies LTE (Long-Term Evolution), IEEE 802.16, IEEE 802.16e, IEEE 802.15.4, IEEE 802.11, Bluetooth does not ensure the security of information flows. In pursuit of super speeds, these channels do not provide confidentiality and integrity services. The Diameter protocol provides interaction

between clients in order to authenticate, authorize and account for various security services, however, it has significant drawbacks in terms of modern cyber attacks. To ensure security in cyber-physical systems based on the Internet of things, the KNX standard (ISO / IEC 14543) is used based on the use of VPN channels (encryption AES-128, -256). However, all security mechanisms will not provide the required level of security in the post-quantum period (the emergence of a full-scale quantum computer). USA NIST experts raise doubts about the strength of modern symmetric and asymmetric cryptosystems (including algorithms on elliptic curves) based on Grover and Shor quantum algorithms. Under such conditions, post-quantum cryptoalgorithms based on the synthesis of theories of error-correcting coding and information protection – crypto-code constructs – can be considered as an alternative security mechanism. Such designs are hybrids, because the formation of an asymmetric cryptosystem (cryptosecurity is not based on a theoretical complexity problem – decoding a random code) is provided on the basis of the use of algebraic codes. According to USA NIST experts, to ensure cryptographic strength, the formation of noise-resistant codes is necessary over the Galois field (GF 2^{10}–2^{13}), which is a rather difficult issue even with modern computing resources. The use in wireless cyber-physical systems requires a significant reduction in the field, which, on the one hand, will ensure a reduction in energy consumption, and on the other hand, it requires the required level of cryptographic strength. Thus, for cyber-physical systems based on wireless mobile technologies, cryptosystems are needed that will provide the required level of cryptographic strength in the post-quantum period, energy intensity, which will allow them to be used in smart technologies, and also provide a full range of security services.

To ensure security in the post-quantum period – the emergence of a full-scale quantum computer, NIST specialists propose to use post-quantum algorithms. Such algorithms require an increase in key sequences to 512 bits for symmetric cryptosystems (this provides a safe time of about 60 years), or the use of post-quantum asymmetric cryptosystems (PQAS). Among the contestants of the third round of the competition, algorithms built on the integration of the theory of error-correcting coding and cryptography stand out. **Fig. 2.16** shows the block diagrams of the McEliece and Niederreiter crypto-code constructions on algebrogeometric codes (elliptic codes over the field GF(2^8), which provide protection against the Sidelnikov attack and reduce energy consumption. In addition, they provide an integrated error correction in the information sequence [36].

However, the McEliece *CCC* provides an integrated (by one mechanism) error correction. The Hamming weight (the number of non-zero elements of the error vector e) does not exceed the correcting ability of the used algebraic block code $\left(0 \le w(e) \le t = \left\lfloor \frac{d-1}{2} \right\rfloor\right)$. The use of *MEC* in crypto-code structures provides the required level of cryptographic strength, due to the use of vectors initialization (IV_i, where i are the numbers of symbols of shortening or lengthening), and also allows to ensure their construction over GF (2^6). The papers [36] present mathematical models and practical algorithms for their implementation, as well as the results of studies of their cryptographic strength. Hybrid crypto-code structures based on defective codes can reduce the level of energy consumption (built over the field GF (2^4), and provide the required level of cryptographic strength, through the use of two-channel cryptography [2, 22, 36]. However, their use

in smart technologies and wireless mobile standards networks is difficult, due to the need the need for additional conversion of m-th code sequences into binary ones and vice versa, which requires additional energy consumption. To solve this issue, it is proposed to use LDPC codes to build crypto-code structures.

LDPC codes are used in modern data transmission standards such as DVB-S2, Gigabit Ethernet, WiMAX, Wi-Fi. This ensures their use in any communication system, for example, in space communications, microwave communication systems, digital satellite television.

The formation of regular LDPC codes is determined by a sequential procedure [36]. A regular LDPC code with block length n is generated based on a check matrix H, which is characterized by a constant number of ones in the row W_r and a constant number of ones in the column W_c. The check matrix H has a low density of units (the density of units is considered low if the specific fraction of units is less than 50 % of all elements of the check matrix).

Based on the given parameters n, W_r, W_c the corrective properties of the code (t bit) are changed. In this case, the position of units in the check matrix H is formed on the basis of random permutations of the columns of the base submatrix containing only one unit in each column. In this case, the rate of a regular LDPC code, depending on the parameters of the check matrix, is determined by the formula:

$$r_k = \frac{n - \left(n \cdot \frac{W_c}{W_r} - \left(W_c - 1 \right) \right)}{n} = 1 - \frac{W_c}{W_r} + \frac{W_c - 1}{n}, \tag{2.12}$$

where n is the length of the code sequence; W_r is the number of ones in the row of the check matrix H; W_c is the number of ones in the column of the check matrix H; r_k is the coding rate of the regular LDPC code.

At the same time, matrices H of the LDPC code of the same size and with the same parameters can generate codes with different code distance d and correction power t.

The check matrix of the LDPC code can be represented as:

$$H = \begin{bmatrix} H_1 \\ \hline \pi_1(H_1) \\ \hline \vdots \\ \hline \pi_{Wc-1}(H_1) \end{bmatrix}, \tag{2.13}$$

where H_1 is the base submatrix; $\pi_1(H_1)$ are the submatrices obtained by random permutation of the columns of the base submatrix H_1, $i = 1, 2, \ldots, W_{c\,1}$.

The check matrix H can be reduced to the form:

$$H = \left[A \middle| I_{n-k} \right], \tag{2.14}$$

where A is some fixed $((n-k)\times k)$ matrix with 0s and 1s (which is no longer 1 – sparse), and I_{n-k} is an identity matrix of size $((n-k)\times(n-k))$.

The codeword generation matrix of G words has the form:

$$G=\left[I_k \middle| -A^T\right]. \tag{2.15}$$

If the matrix H is presented in the form (2.14), then the matrix G (2.15) is easily obtained from the matrix H by Gaussian transformations.

On the receiving side, an authorized user who knows the concealment matrices uses a fast algorithm based on soft decoding. **Fig. 2.16** is a block diagram of decoding the received sequence based on soft decoding.

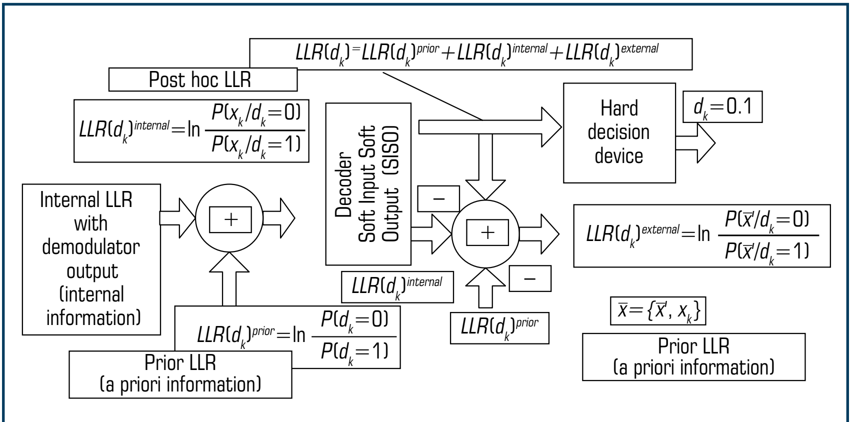


Fig. 2.16 Soft Decision Decoding Scheme

The following designations are introduced on the scheme: *LLR* – *log-likelihood ratio* (logarithm of the likelihood ratio); d_k – a codeword symbol, $d_{ij}\in\{0, 1\}$, $x_k=(2d_k1)+p_k$; p_k – a random variable with a normal distribution with zero mean.

Analysis of **Fig. 2.16** shows that the soft decision is the log likelihood ratio (posterior LLR). A soft decision can be represented by a set of a priori, internal and external information. The hard decision for some symbol is based on the a posteriori LLR. The sign of the logarithm with respect to the likelihood determines the rigid solution, and the magnitude determines the reliability of this solution.

The check matrix has the dimension $(N-K)\times N$ and allows expressing $(N-K)$ check symbols P_1, P_2, ..., P_{N-K} in the form of a linear combination of information symbols d_k, $k=1, 2, \ldots, K$, that is, it defines the check equations:

$$\begin{cases} P_1 = c_{11}d_1 \oplus c_{21}d_2 \oplus \ldots \oplus c_{k1}d_K \\ P_2 = c_{12}d_1 \oplus c_{22}d_2 \oplus \ldots \oplus c_{k2}d_K \\ \ldots \\ P_{N-K} = c_{1\,N-K}d_1 \oplus c_{2\,N-K}d_2 \oplus \ldots \oplus c_{K\,N-K}d_K \end{cases}, \tag{2.16}$$

where c_{ij} are elements of submatrix A, $c_{ij} \in \{0,1\}$. If d_k, $k = 1, 2, \ldots, K$ are statistically independent symbols that take the values 0 and 1 and which, in general, correspond to the information symbols of the block code, and $\beta_k = (2d_k\text{-}1) = \pm 1$.

With this format, the result of adding symbols β_k modulo two will look like this:

$$\begin{cases} \beta_1 \oplus \beta_2 = -1, \text{ if } \beta_1 = \beta_2 \\ \beta_1 \oplus \beta_2 = +1, \text{ if } \beta_1 \neq \beta_2 \end{cases}. \tag{2.17}$$

Then the logarithm of the likelihood ratio of the sum modulo – two symbols β_k – $LLP(\beta_1 \oplus \beta_2 \oplus \ldots \oplus \beta_K)$ can be written as follows [60]:

$$LLR(\beta_1 \oplus \beta_2 \ldots \oplus \beta_k \oplus \ldots \oplus \beta_K) = \ln\left[\frac{\sum_k e^{LLR(\beta_k)}}{1 + \prod_k e^{LLR(\beta_k)}}\right]. \tag{2.18}$$

Expression (11) can be approximated as:

$$LLR(\beta_1 \oplus \beta_2 \ldots \oplus \beta_k \oplus \ldots \oplus \beta_K) \approx -1 \cdot \left[\prod_k \text{sign}\{LLR(\beta_k)\}\right]\left[\min_k\left\{\left|LLR(\beta_k)\right|\right\}\right], \tag{2.19}$$

where the *sign* (•) function returns the sign of its argument.

Each test equation (9) allows to express one symbol (regardless of whether it is information or test) through the modulo two sum of all other symbols included in this test equation.

The initial data of the algorithm are: a check matrix H of a block (N, K) code, a sequence of soft decisions for information and check symbols from the demodulator output.

Decoding algorithm for LDPC codes:

Step 1. Determine the reliability estimate for each code symbol (for each information and check symbol) of the code word based on soft decisions on the demodulator output by calculating their absolute value (neglect the sign of soft decisions in the demodulator output sequence).

Step 2. For the row of the check matrix H with number i, $i = 1 \ldots N{-}K$:

1) find the code symbol corresponding to the non-zero (single) value of the elements of the row with number i of the matrix H. This means that the code symbol is part of the verification equation determined by the row with number i, and has the lowest reliability score value (the least reliable

symbol). Let's fix the column number j, $j = 1 \ldots N$ of the check matrix H, which corresponds to the found least reliable character;

2) transform the check matrix H by linearly combining its rows. Linear combination is performed in order to eliminate the dependence of other test equations (defined by other rows of the test matrix) on the least reliable symbol found. This will be achieved when the column of the matrix H with the number j will have only one unit, contained exactly in the considered row with the number i;

3) repeat preliminary procedures 1 and 2 for each of the rows of the check matrix H, after which let's proceed to the next step.

Step 3. Perform hard decoding of the K symbols that have the highest value for reliability estimation (the most reliable symbols).

Step 4. For each of the K most reliable symbols:

1) find soft solutions using two trial code sequences (hypotheses). One trial sequence is generated by re-encoding the hard decoding result of the K most reliable symbols obtained in Step 3 (first hypothesis). The other is formed by re-encoding the result of hard decoding of the K most reliable symbols obtained in step 3, but with an additional inversion of the symbol for which a soft solution is found (second hypothesis);

2) find a hard solution based on the soft solution obtained in the preliminary procedure.

Step 5 (optional). Update the reliability estimate for each code symbol and proceed to Step 1 for the next iteration.

Thus, the presented algorithm makes it possible to ensure the efficiency of decoding and ensures the use of LDPC codes in McEliece and Niederreiter crypto-code structures. This approach allows varying, depending on the degree of information secrecy in the choice of an error-correcting code for crypto-code structures, and ensuring the required level of security.

Thus, taking into account expressions (2.12)–(2.15) and structural schemes of *CCC* McEliece and Niederreiter, it is possible to use post-quantum cryptosystems of provable strength to ensure information security in wireless networks based on mobile technologies [36].

To ensure security in cyber-physical systems and smart technologies, the KNX standard (ISO / IEC 14543) is usually used, which provides security services – data confidentiality and integrity [2, 22, 36]. **Fig. 2.17, 2.18** present the basic principles of security based on the use of the KNX standard. The KNX IP Secure standard allows authentication and encryption of KNX telegrams in IP networks. In this case, as a rule, tunneling is formed, which ensures the confidentiality of information. KNX IP Secure mechanisms are an additional security shell (shell) that protects all KNXnet / IP data traffic.

However, KNX IP Secure is not so secure, it is possible to monitor the network, record sent packets and easily repeat them, because there are no line connectors with the "Security Proxy" function. In addition, the use of the AES-128 algorithm in the formation of tunneling in the post-quantum period will not provide the required level of protection even for the inner loop.

In **Fig. 2.18**, the presented interaction protocol based on wireless mobile Internet confirms the possibility of ensuring data confidentiality only in the internal security loop – inside the network

infrastructure. However, in the outer security loop, the standard does not provide services. It is assumed that this is done by security technologies inside cloud platforms, which, given their accessibility to the intelligence services of developed countries, casts doubt on the provision of security services. Thus, a control system that is hosted and implemented on the basis of cloud technologies (an external security loop) is not fully secure. With the advent of quantum computers, the possibility of performing the full range of functions with safety in mind is called into question.

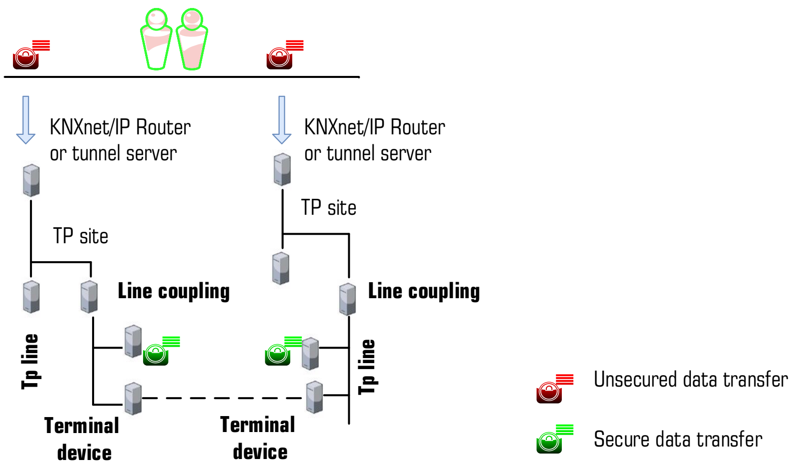


Fig. 2.17 Ensuring security in mobile wireless channels based on KNX

KNX Data Secure allows to protect user data from unauthorized access and manipulation using encryption and authentication mechanisms. KNX Data Secure devices use a longer KNX telegram format (extended frames) than conventional devices to transmit authenticated and encrypted data.

KNX Data Secure uses CCM (Cipher Chain Message Authentication Code Counter) mode with 128-bit AES encryption to ensure the integrity of the information. However, the proposed options for using the KNX standard provide only integrity and do not provide confidentiality of information, which significantly reduces the overall level of security of information flows circulating in wireless mobile networks.

To ensure authenticity in mobile wireless technologies and networks, the Diameter protocol is used. The Diameter protocol has a predefined set of common attributes and assigns appropriate semantics to each attribute. These AVPs (Attribute-Value-Pair) convey AAA (authentication, authorization, accounting) details (such as routing, security, and capabilities) between two Diameter

nodes. In addition, each AVP pair is associated with an AVP Data Format that is defined in the Diameter protocol (e.g., OctetString, Integer32), so the value of each attribute must follow the data format [2, 22, 36]. However, the Diameter protocol, like previous mobile network protocols, was not designed with security in mind. Therefore, it has almost all the threats that the "G" technology itself has.

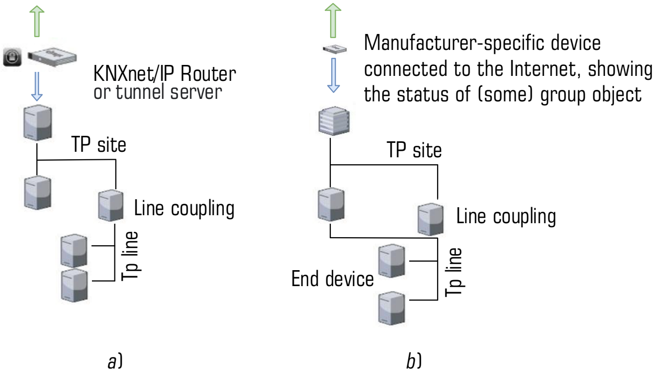


Fig. 2.18 KNX Data Secure: *a* – KNX IP Secure; *b* – KNX Data Secure

Developers in pursuit of super-speeds do not think that the development of computer technology allows intruders (cyberterrorists) to "expand the range and boundaries" of threats. In other words, to consider the use of this technology for organizing a "window" to corporate networks and / or local user networks.

As practice shows, in networks based on the Diameter protocol, attacks aimed at denial of service, disclosure of information about subscribers and the operator's network, as well as fraud against the operator are possible.

In addition, an attacker can forcefully transfer the subscriber's device to 3G mode – and carry out further attacks on the less secure SS7 system.

The goals of attacks are listening to voice calls, intercepting SMS, and implementing fraudulent schemes against subscribers [22, 36]. Thus, the absence of cryptographic algorithms for providing confidentiality and integrity services leads to the identification of the following classes of "classical" attacks (**Fig. 2.19**).

At the same time, confidentiality means protecting data from passive attacks during transmission, integrity means protecting data during storage, and authenticity means the authenticity of the message source.

Analysis of **Fig. 2.19** shows that if only this protocol is present in mobile wireless technologies, the problems of confidentiality and integrity are not solved. The use of KNX standard mechanisms provides these services only inside the infrastructure of cyber-physical systems, and does not provide protection in the external security loop – a platform based on cloud technologies. In **Table 2.1** shows the main characteristics of wireless mobile and computer networks and security services based on the KNX standard and the Diameter protocol.

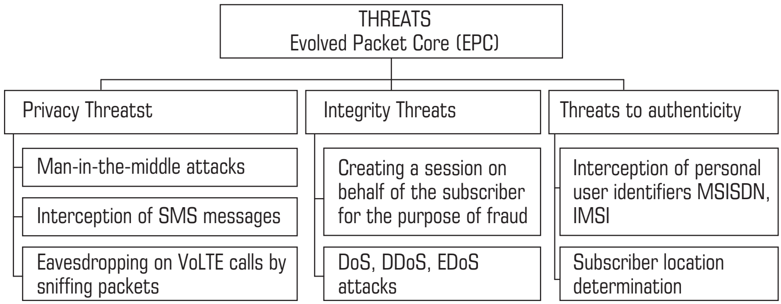


Fig. 2.19 Main types of attacks on Evolved Packet Core

Examples of the practical implementation of such systems is the protocol for ensuring the security of voice messages in online mode, proposed in [36], based on McEliece and Niederreiter *CCC* on ***EC*** (***MEC***), which is shown in **Fig. 2.20**.

So, in **Fig. 2.21** to ensure the security of voice messages, it is proposed to use a hardware-software encoder that is built into the headset headset (it is proposed to use Bluetooth headphones) and provides encryption of a digital message based on the McEliece *CCC*. After that, the encrypted message is transmitted via a Bluetooth channel to a mobile gadget. In this case, standard protocols of the GSM mobile Internet channel are used. This allows to ensure the confidentiality of the conversation without taking into account the requirements of the communication channel, the requirements of manufacturers of headsets and mobile gadgets, and does not take into account modifications of both the Bluetooth channel and mobile Internet technology.

In addition, the use of a hardware-software implementation of the encoder in the form of a chipset can significantly reduce the cost of production and implementation of this approach. To ensure security, only the session password is recorded in the headphones, depending on the role (sender, recipient), which are recorded from the mobile application.

After the end of the conversation, they are deleted. In this case, the chipset implements an encoder based on the McEliece *CCC*. Ensuring the security of the transfer of key data between the

mobile application and the server is provided by Niederreiter's *CCC*. To ensure the security of the server part, after generating the keys for conducting a conversation and transferring them to the sender and recipient, the server RAM is reset, which ensures channel tunneling between users. The secret keys of the McEliece and Niederreiter crypto-code structures change at different time intervals and are OTP keys (session keys).

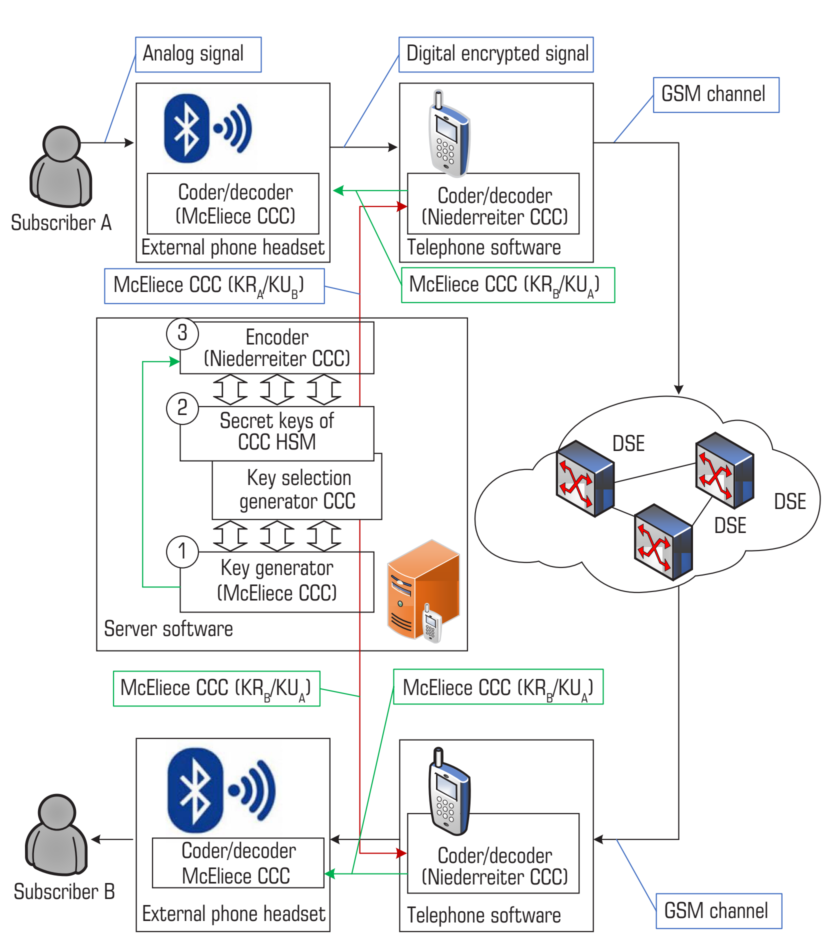


Fig. 2.20 Structural diagram of building a two-loop information protection system on the CCC to ensure the confidentiality of voice messages

Consider a voice message security protocol based on post-quantum algorithms:

SUBSCRIBER A (call initiator)

1. Opens the phone software and in the list of subscribers find the corresponding subscriber (Subscriber B).

2. Sends a request to subscriber B through the server.

3. Receives on the phone software through a private channel (using encryption based on Niederreiter's CCC on the *EC*) a private key, and a public key of Subscriber B.

4. Confirms readiness for a conversation. At the same time, a private key KR_A and a public key KU_B are transmitted from the phone software via a Bluetooth channel.

5. In the Bluetooth headphones in the encoder (coder / decoder), the key is recorded.

6. After the key is written, a ready signal is generated.

7. After confirmation of the readiness of subscriber B, a conversation is carried out.

SERVER SOFTWARE

1. At the request of subscriber A in Secret keys of CCC (block 2), the *CCC* Key Selection Generator randomly selects the key parameters and sends it to key generator (block 1).

2. In key generator, secret keys are received from GSM (masking matrices – X, P, D, and generating matrix G^{EC}).

3. In key generator, KR_A (McEliece *CCC* private key of subscriber A) and KU_A (public key of Subscriber A) are formed.

4. Based on the response of subscriber B, a public key KU_B is formed and transmitted to Subscriber A.

5. In encoder (block 3) from key generator (bock 1), the generated KR_A and KU_A are received, after the transfer of the keys in key generator, the data is erased.

6. In encoder KR_A, KU_A, KU_B are encrypted.

7. From encoder, respectively, KR_A, KU_B are sent to Subscriber A (the subscriber who initiates the call), KU_A – to Subscriber B (the subscriber who is being called), after the transfer of the keys in encoder, the data is erased.

SUBSCRIBER B (recipient of the call)

1. Receives a request from the server in the phone software to transfer the public key (KU_A).

2. Confirms the request to the server, sends KR_B.

3. Receives the public key KU_A on the phone software via a private channel (using encryption based on Niederreiter CCC in the EC).

4. Confirms readiness for a conversation. At the same time, a public key (KU_A) is transmitted from the phone software via a Bluetooth channel.

5. In the Bluetooth headphones in the decoder (coder / decoder), the key is written.

6. After the key is written, a ready signal is generated.

7. After confirmation of readiness, subscriber B sends a signal to the server that it is ready to talk.

Thus, the proposed protocol ensures the closure of the mobile Internet channel using a set of software and hardware. Using a hardware solution for closing (encrypting) a voice message in a

headset will counteract almost all threats, and using a key server provides a tunnel mode, which eliminates the possibility of "eavesdropping" of voice messages.

Fig. 2.21 shows the practical implementation of the proposed Concept and crypto-code structures on LDPC codes. The proposed protocol for ensuring security in cyber-physical systems ("Smart Home") is based on the use of a two-loop security concept and post-quantum algorithms.

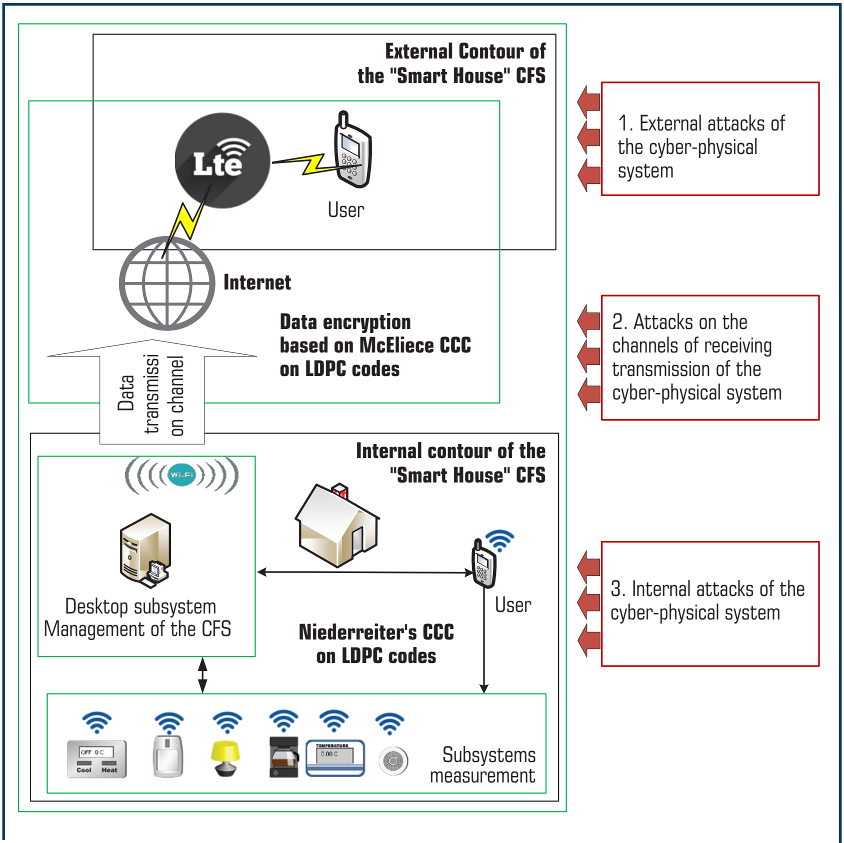


Fig. 2.21 Structural diagram for constructing a two-circuit protection system of the "Smart House" system based on CCC

The use of these post-quantum asymmetric cryptosystems will provide the required level of security when providing security services. The use of LDPC codes allows using mobile wireless technologies based on IEEE 802.11ac, IEEE 802.11ax, IEEE 802.16m, IEEE 802.15.1, IEEE 802.15.4

standards without significant changes. The smart home system controls a complex of autonomous systems, each of which controls certain devices in the house, connecting their common cyber-physical system. However, to ensure the security of the external circuit (control and information storage systems), it is proposed to use the developed server, which is physically located in the house.

Each system sends a data packet to a local server, which allows to manage your home in the absence of the Internet, being on the same local network (being connected to a Wi-Fi router). Information in the network of the cyber-physical system is transmitted over open wireless channels with encryption based on the McEliece and Niederreiter *CCC* using LDPC codes.

This approach provides security services, and through the use of a local control server, it reduces the likelihood of targeted attacks to gain unauthorized access to the Smart Home control system. Also, the approach provides the required level of security when using mobile control applications, based on the use of *CCC* McEliece and Niederreiter on LDPC codes.

To ensure the security of the database, McEliece and Niederreiter *CCC* on the **EC** (**MEC**) can be used, which will greatly complicate the possibility of implementing cyber attacks of the R2L class (Remote to Local (user) Attack).

2.6 MATHEMATICAL MODEL FORMATION OF A POST-QUANTUM ALGORITHM UMAC

To ensure the formation of MAC codes that meet the requirements of stability (security) under the conditions of quantum cyber threats, the efficiency of transformations and the preservation of the universality property, it is proposed to use an improved UMAC algorithm. The main advantages of the proposed approach are shown in **Fig. 2.22**. The basis of the modification is the use of crypto-code constructions with algebrogeometric and / or flawed codes to form a pseudo-substrate on the third layer of the UMAC hashing algorithm. This approach provides guaranteed stability in the post-quantum cryptoperiod, the use of the fastest possible transformations of the MAC code formation, and the preservation of universality properties. This also allows expanding the range of use of the proposed approach. Let's consider the mathematical apparatus for generating a hash code based on the improved UMAC algorithm on crypto-code constructions with algebrogeometric codes.

Input data for the mathematical model of hash code generation

To build mathematical models for the formation of a hash code of a transmitted message and a pseudo-random background, the following input data are used:

M – transmitted plaintext;

I – plaintext information symbols (*k*-dimentional information vector over $GF(q)$);

K – secret key;

Taglen – an integer from a set of valid values that specifies the length of the Tag message authentication code in bytes;

Hash (K, M, Taglen) – the function of the key universal hashing of the informational message using the secret key *K*;

Y_{L1l} – value of the universal hash function (UHASH-hash) of the first level of hashing;

Y_{L3l} – hash value (Carter-Wegman-hash) of the third level of hashing;

T – data block;

Blocklen – data block length (bytes);

Keylen – secret key length (32 bytes);

Tag – integrity and authenticity control code;

K_{L1l} – secret key of the first hashing level, consisting of subkeys $K_1, K_2, \ldots, K_n$;

K_{L3l} – secret key of the second level of hashing, consisting of keys K_{L31} (subkeys $K_1, K_2, \ldots, K_n$) and K_{L32} (subkeys $K_1, K_2, \ldots, K_n$);

Numbyte – pseudo-random key sequence length (number of subkeys);

K' – pseudo-random key sequence;

Index – number of subkey;

$Wordbits \in [64, 128]$;

Maxwordrange – positive integer less than $2^{Wordbits}$;

k – key K_{L2} dependent integer from range $[0, \ldots, prime(Wordbits) - 1]$, $prime(x)$ – largest prime number less than 2^x;

$M_P = Y_{L1} = Hash_{L1}(K_{L1}, M)$ – data to be polynomial hashed.

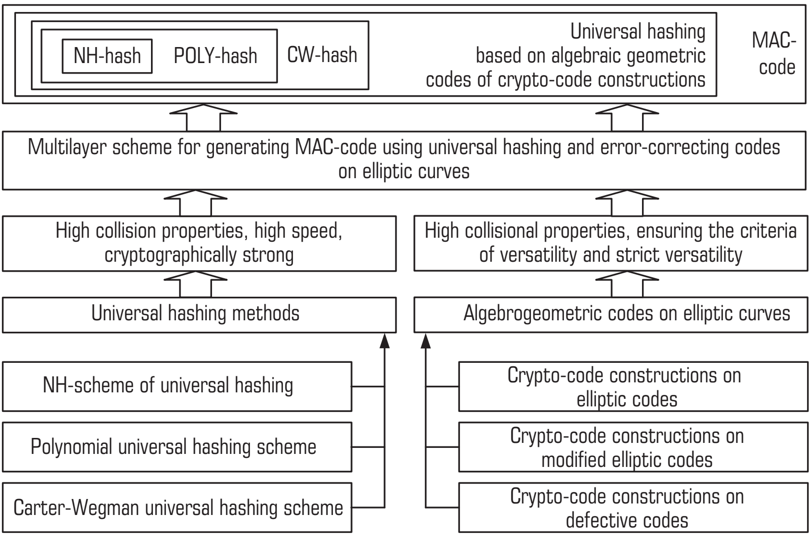


Fig. 2.22 Scheme of cascading generation of data integrity and authenticity control codes using the UMAC algorithm on the CCC

Using this methodology, it is proposed to check hash codes generated using the modified UMAC algorithm. A formalized presentation of this methodology allows to develop a practical algorithm for its implementation. Let`s consider schemes for checking hash codes for the possibility of collisions based on universality and strict universality criteria.

The algorithm for checking hash codes to comply with the rules of the universal class of hash functions is shown in **Fig. 2.23**.

Mathematical model of hash code generation

Based on the cascade representation of the UMAC algorithm, the mathematical model for generating the hash code of the transmitted message will consist of three levels.

First level hash code performs an array-string of M dimention split to 2^{64} bytes to blocks M_i 1024 bytes each followed by the transformation of each block by the function $NH(K_{L1}, M_i)$. Obtained results $Hash_{L1_i} = NH(K_{L1}, M_i)$ concatenated (combined) into a string $Y_{L1} = Hash_{L1}(K_{L1}, M)$, which is 128 times shorter than the information sequence.

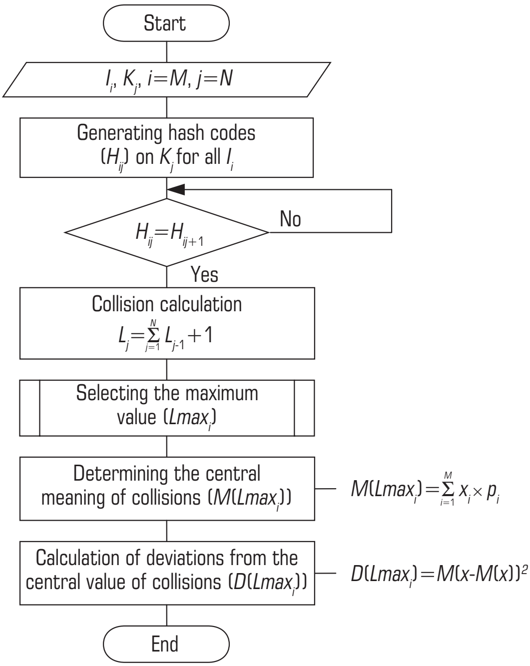


Fig. 2.23 Algorithm for checking hash codes to meet the requirements of the universal class of hash functions

This string is the first level hash result:

$$Y_{L1} = Hash_{L1}\left(K_{L1}, M\right) = NH\left(K_{L1}, M_0\right) \| NH\left(K_{L1}, M_1\right) \| \ldots \| NH\left(K_{L1}, M_{n-1}\right),$$

where $n=[Length(M)/1024]$, $[x]$ – the integer part of number x, $Length(M)$ – byte length of information message of length M.

Function $Hash_{L1_i} = NH\left(K_{L1}, M_i\right)$ value is calculated according to the following rule. Information block M_i is split into four-byte subblocks so that

$$M_i = M_{i_1} \| M_{i_2} \| \ldots \| M_{i_t},$$

where $t=[Length(M_i)/4]$.

In this case $t=[1024/4]=256$.

Similarly, the key sequence K_{L1} represented as sequences of four-byte subblocks:

$$K_{L1} = K_{L1_1} \| K_{L1_2} \| \ldots \| K_{L1_t}.$$

After that (taking the initial state $Hash_{L1_i} = 0$) for all $j = 1, 9, 17 \ldots, t-7$ the following operations are performed:

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+0}} +_{32} K_{L1_{j+0}}) \times_{64} (M_{i_{j+4}} +_{32} K_{L1_{j+4}})),$$

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+1}} +_{32} K_{L1_{j+1}}) \times_{64} (M_{i_{j+5}} +_{32} K_{L1_{j+5}})),$$

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+2}} +_{32} K_{L1_{j+2}}) \times_{64} (M_{i_{j+6}} +_{32} K_{L1_{j+6}})),$$

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+3}} +_{32} K_{L1_{j+3}}) \times_{64} (M_{i_{j+7}} +_{32} K_{L1_{j+7}})),$$

where $+_{64}$, $+_{32}$ – modulo addition operations 2^{64} and 2^{32}, respectively; $\times_{64}$ – modulo multiplication operation 2^{64}.

As a result of calculations, an eight-byte value is formed $Hash_{L1_i}$.

Second level hash code uses polynomial key transformation $Poly$. The result of this level is getting a hash code $Y_{L2} = Hash_{L2}\left(K_{L2}, Y_{L1}\right) = Poly(Wordbits, Maxwordrange, k, M_P)$, that is, the second level hash input is a string $Y_{L1} = Hash_{L1}\left(K_{L1}, M\right)$.

As input, the polynomial hash function uses:

According to the specification of the UMAC algorithm as $prime(x)$ the following constants are used: $prime(36) = 2^{36} - 5$, $prime(64) = 2^{64} - 59$, $prime(128) = 2^{128} - 159$. Bit length M_P denoted $Bytelength(M_P)$. Depending on the length M_P the following features are used in the implementation of the second level of hashing:

– if the length of the received data M_P does not exceed 2^{17} bytes, then polynomial hashing $Poly$ executed with parameters $Wordbits = 64$; $Maxwordrange = 2^{64} - 2^{32}$; $k = k64$ – the string formed by the first eight bytes of the key K_{L2} and a special eight-byte mask;

– if the length of the received data M_P surpasses 2^{17} bytes (but does not exceed 2^{64} bytes), then the first 2^{17} bytes of data are processed by the polynomial hash function $Poly(64, 2^{64} - 2^{32}, k64, M_P)$, and the remaining data bytes are processed by the function $Poly$ with parameters $Wordbits = 128$; $Maxwordrange = 2^{128} - 2^{96}$; $k = k128$ – the string formed by the last 16 bytes of the key K_{L2} and a special 16 byte mask.

Hashed data M_P broken down into blocks $Wordbytes = Wordbits / 8$ bytes:

$$M_P = M_{P_1} \| M_{P_2} \| \ldots \| M_{P_n},$$

where $n = Bytelength(M_P) / Wordbytes$.

The result of hashing is the value of the polynomial function

$$C_j = M_i \times G_{X_{a_j}}^{ECCu^T} + e,$$

which is calculated by an iterative procedure (for all $i = 1, 2, \ldots, n$):

$$Poly_i = \left(kPoly_{i-1} + M_{P_i}\right) \mathrm{mod}(p), \quad Poly_0 = 1, \quad p = prime(Wordbits),$$

using Horner's scheme:

$$M_{P_n} + kM_{P_{n-1}} + \ldots + k^{n-1}M_{P_1} + k^n = (((k + M_{P_1})k + M_{P_2})k + \ldots + M_{P_{n-1}})k + M_{P_n}.$$

The calculated hash value $Y_{L2} = Poly_n$ is an integer from the range $[0, \ldots, prime(Wordbits) - 1]$.

Third level hash code $Hash_{L3}(K_{L3_1}, K_{L3_2}, Y_{L2})$ is performed on the result of polynomial hashing and converts data of length up to 16 bytes given to its input into a hash code *Y* of a fixed length 32 bits.

The initial data of the third level of hashing are two key sequences K_{L3_1} and K_{L3_2} of lengths 64 and 4 bytes, respectively, as well as an input 16 bytes sequence Y_{L2}.

Hashed data Y_{L2} and key sequence K_{L3_1} are evenly divided into eight blocks, each of which is represented as an integer Y_{L2_i} and $K_{L3_{1i}}$, $i = 1, 2, \ldots, 8$.

Hash value Y_{L3} is calculated as follows:

$$Y_{L3} = \left(\left(\left(\sum_{i=1}^{m} Y_{L2_i} K_{L3_{1i}}\right) \mathrm{mod}(prime(36))\right) \mathrm{mod}(2^{32})\right) xor(K_{L3_2}),$$

where $(x)xor(y)$ – the XOR operation on values *x* and *y*.

A mathematical model of encoding and decoding information when transmitting and receiving via communication channels based on the McEliece scheme using shortened and elongated modified elliptic codes (MEC)

When transmitting a message, a modified UMAC algorithm is used to provide plaintext authentication, and a crypto-code structure based on a McEliece scheme using modified elliptic codes (MEC) is used as key data generation.

It is necessary to take into account that this model of information transformation in the McEliece system can be built as on shortened ($IV_1 = EC–h_i$) [3], and on elongated (IV_1, $IV_2 = EC–h_r$) [2, 22] MEC. Additional initialization vectors provide "compensation" for the decrease in the power of the alphabet during the formation of the McEliece CCC on the MEC. This approach ensures the preservation of versatility and provides stability in post-quantum cryptography.

The mathematical model of the crypto-code construction (CCC) on the McEliece scheme using modified elliptic codes (MEC) based on shortening (reduction of information symbols) / extension (adding information symbols) is formally defined by the set of elements listed below [2, 22] in **Table 2.12**.

Regardless of the way the code is presented (for both shortened and elongated versions), let's set a number of model parameters as follows.

So, in the CCC based on the McEliece scheme on a modified (shortened / elongated) algebraic (n, k, d)-code with a fast decoding algorithm that "disguises" as random (n, k, d)-code by multiplying the generating matrix G^{EC_i} code for secret masking matrices X, P and D [2, 36], authorized user public key generation is provided:

$$G_X^{EC_i} = X^i \times G^{EC_i} \times P^i \times D^i. \quad (2.20)$$

Formation of a closed text c_X open message M for a given public key can be represented as follows [2, 36]:

$$c_X = i \times G_X^{EC_i} + e, \quad (2.21)$$

where i is information package for forming a code word, e is randomly generated vector e (e_0, e_1, ..., e_{n-1}).

After the generated codogram, it is shortened or lengthened using additional session keys of initialization vectors IV_1, IV_2 (depending on the modification algorithm used).

When generating key data in the proposed UMAC algorithm, it is proposed to use the generated cryptogram of CCC McEliece on the MEC, which is used as input to generate the secret key K (with length *Keylen*) and random number *Nonce* eight bytes in size. When specifying the size of the generated hash code *Tag* integer *Taglen* used by formula [2]:

$$Tag = Hash\ (K,M,Taglen) \oplus PDF(K,Nonce,Taglen).$$

Table 2.12 Definition of model elements

<table>
<tr><th rowspan="2">№</th><th rowspan="2">Parameter name</th><th colspan="2">Formal description of the parameter taking into account</th></tr>
<tr><th>character shortening</th><th>character extensions</th></tr>
<tr><td>1</td><td>Plaintexts set</td><td>$M=\{M_1, M_2,\dots, M_{q^k}\}$, where $M_i=\{I_0, I_{h_1},\dots, I_{h_j}, I_{k-1}\}$, $\forall I_j \in GF(q)$, h_j – information symbols equal to zero, $|h| = 1/2k$, that is $I_i = 0$, $\forall I_i \in h$</td><td>$M=\{M_1, M_2,\dots, M_{q^k}\}$, where $M_i=\{I_0, I_{h_1},\dots, I_{h_j}, I_{k-1}\}$, $\forall I_j \in GF(q)$, h_j – information symbols equal to zero, $|h| = 1/2k$, that is $I_i = 0$, $\forall I_i \in h$, h_r – extension information symbols k, $|h| = 1/2k$</td></tr>
<tr><td>2</td><td>Closed texts (codograms) set</td><td>$C=\{C_1, C_2,\dots, C_{q^k}\}$, where $C_i=(A^*_{X_0}, A^*_{h_1},\dots,A^*_{h_j},A^*_{X_{n-1}})$, $\forall I_j\in GF(q)$</td><td>$C=\{C_1, C_2,\dots, C_{q^k}\}$, where $C_i=(A^*_{X_0}, A^*_{h_1},\dots,A^*_{h_j},A^*_{X_{n-1}})$, $\forall I_j\in GF(q)$</td></tr>
<tr><td>3</td><td>Direct mappings (based on the use of the public key – the generating matrix) set</td><td>$\varphi=\{\varphi_1,\varphi_2,\dots,\varphi_s\}$, where $\varphi_i: M\to C_{k-h_r}$, $i=1,2,\dots,s$</td><td>$\varphi=\{\varphi_1,\varphi_2,\dots,\varphi_s\}$, where $\varphi_i: M\to C_{k-h_r}$, $i=1,2,\dots,s$</td></tr>
<tr><td>4</td><td>Inverse mappings (based on the use of a private (private) key – masking matrices) set</td><td>$\phi^{-1}=\{\phi_1^{-1},\phi_2^{-1},\dots,\phi_s^{-1}\}$, where $\phi_i^{-1}: C_{k-h_r}\to M$, $i=1,2,\dots,s$</td><td>$\phi^{-1}=\{\phi_1^{-1},\phi_2^{-1},\dots,\phi_s^{-1}\}$, where $\phi_i^{-1}: C_{k-h_r}\to M$, $i=1,2,\dots,s$</td></tr>
<tr><td>5</td><td>A set of keys parameterizing direct mappings (public key of an authorized user)</td><td colspan="2">$K_{a_i}=\{K_{1_{a_i}},K_{2_{a_i}},\dots,K_{s_{a_i}}\}=\{G^{EC_1}_{X\ a_i}, G^{EC_2}_{X\ a_i},\dots,G^{EC_s}_{X\ a_i}\}$,
where $G^{ECi}_{X\ a_i}$ – generating $n\times k$ matrix of disguised as a random code algebraic geometric block (n, k, d)-code with elements from $GF(q)$, a_i – set of coefficients of a polynomial of a curve $a_1\dots a_6$, $\forall a_i\in GF(q)$, defining a specific set of curve points from space P^2
$\varphi_i: M \xrightarrow{K_{ia_i}} C_{k-h_i}$, $i=1,2,\dots,s$</td></tr>
<tr><td>6</td><td>A set of keys that parameterize reverse mappings (private (closed) key of an authorized user)</td><td colspan="2">$K^* = \{K_1^*,K_2^*,\dots,K_s^*\}=\{\{X,P,D\}_1,\{X,P,D\}_2,\dots,\{X,P,D\}_s\}$,
$\{X,P,D\}_i=\{X^i,P^i,D^i\}$,
where X^i – a masking non-degenerate randomly uniformly formed by a key source $k\times k$ matrix with elements from $GF(q)$, P^i – permutation randomly uniformly generated by the key source $n\times n$ matrix with elements from $GF(q)$, D^i – diagonal formed by key source $n\times n$ matrix with elements from $GF(q)$, that is $\varphi_i^{-1}: C\xrightarrow{K_i} M$, $i=1,2,\dots,s$</td></tr>
</table>

This authentication code will be part of the hash code of the open message M. Open message M and closed text c_X hashing processes occurs according to the same procedure for generating UMAC tags using the universal hash function UHASH (K, M, Taglen) [2].

Let's consider the UMAC algorithm based on the UHASH function, which is formed in three stages. At the first stage, UHASH-hash is applied to the input message, at the second stage –

POLY-hash is applied to this result, and finally, at the third stage, Carter-Wegman-hash is applied to the result. If the length of the input message is no more than 1024 bits, then POLY-hash is not used. Since the Carter-Wegman-hash function returns only a word of 4 bytes in length, then if it is needed to get a hash of more than 4 bytes in length, several iterations of this three-level scheme are performed (**Table 2.13**).

Table 2.13 Obtaining hash codes for open messages

Hashing stages	Hashing process	Hash result by sending and receiving parties
UHASH-hash	Splitting a message into blocks of 1024 bytes, receiving a message 128 times smaller than the input	$Y_{L1} = Hash_{L1}(K_{L1}, M)$ $Y'_{L1} = Hash'_{L1}(K'_{L1}, M)$
POLY-hash	Verification of data integrity and authenticity of the message, receiving a 16-byte number	$Y_{L2} = Hash_{L2}(K_{L2}, Y_{L1})$ $Y'_{L2} = Hash'_{L2}(K'_{L2}, Y'_{L1})$
Carter-Wegman-hash	Getting a 4-byte value from a 16-byte number	$Y_{L3} = Hash_{L3}(K_{L31}, K_{L32}, Y_{L2})$ $Y'_{L3} = Hash'_{L3}(K'_{L31}, K'_{L32}, Y'_{L2})$

When transmitting a message to the communication channel, the result of concatenation of three components: the information message M, cryptogram c_X and hash code Y (**Fig. 2.24**).

On the receiving side, to verify the integrity of the received message, an authorized user, knowing the initialization vectors IV_1, IV_2 performs the following actions (**Fig. 2.25**):

– based on the Berlekamp-Messi fast decoding algorithm, an error vector is found;

– the resulting error vector is used to generate a codogram using the CCC McEliece at MEC;

– verification of the received and generated codogram is carried out on the basis of the CCC McEliece at MEC. If the codograms (cryptograms) do not match, the message is considered modified and a request for re-sending from the sender is generated;

– when the codograms match, a hash code Y' is created from closed text c_X' according to the scheme given in **Table 2.13**;

– verification of the received and generated hash code is carried out. If the hash codes do not match, the message is considered modified and a request for re-sending from the sender is generated.

During the formation of the circuit, the following elements were also used as input data:

M – plain text;

G – user private key;

X – non-degenerate k×k matrix over $GF(q)$;

P – permutation n×n matrix over $GF(q)$;

D – diagonal n×n matrix over $GF(q)$;

H – transposed matrix based on test r×n matrix of elliptic code over $GF(q)$.

Double verification allows to provide a high level of integrity and reliability of the transmitted message, while the algorithm used provides a high level of speed and cryptographic stability of the hash code in post-quantum cryptography.

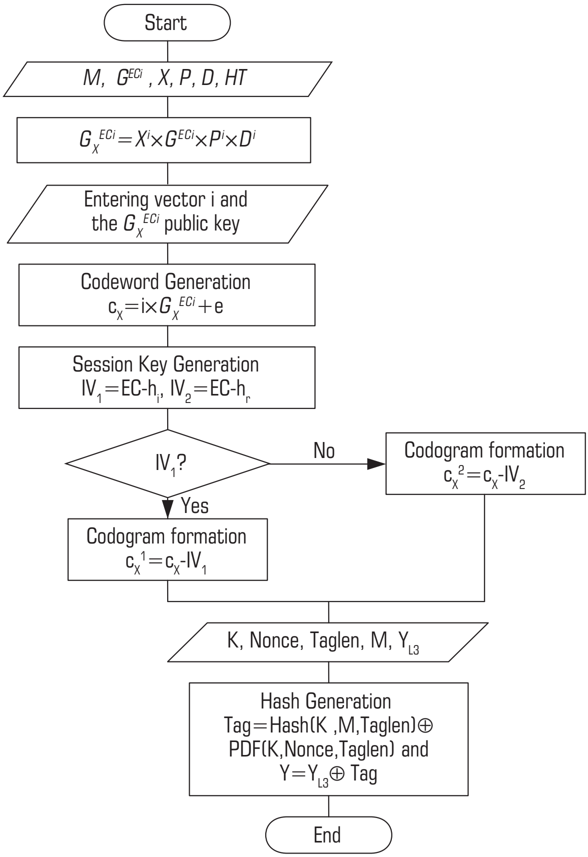
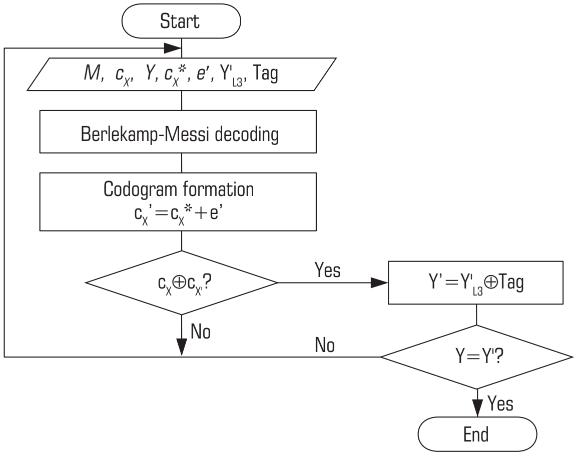
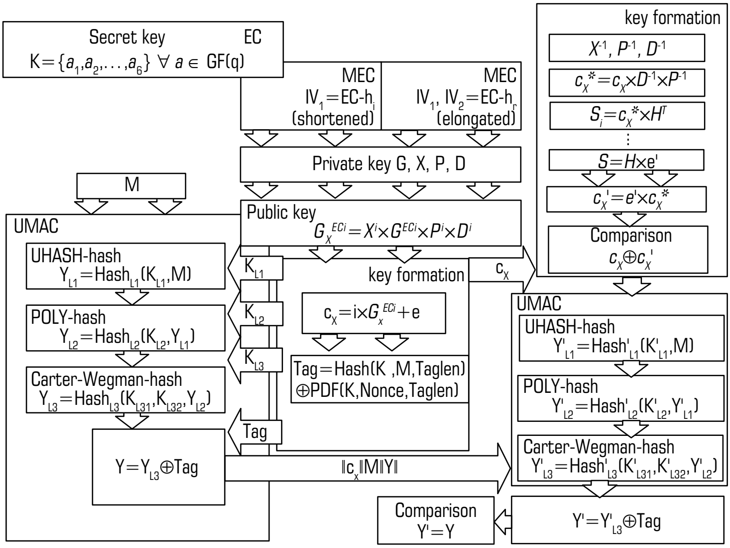


Fig. 2.24 Formation of an algorithm for encrypting a sender message using a CCC based on McEliece at MEC

Schematically, the process of confirming the integrity of information during transmission from the sending to the receiving side based on verification of codograms and hash codes using the CCC McEliece at MEC is presented in **Fig. 2.26**.



⬡ **Fig. 2.25** The algorithm for checking the integrity of the received message



⬡ **Fig. 2.26** The scheme of transmitting a message from the sender to the recipient and checking the integrity of the received through a comparison of the codograms and hash codes using the CCC McEliece at MEC

Mathematical model of the formation of a pseudo-random substrate *Pad* based on the McEliece HCCC

Input data for the mathematical model for the formation of a pseudo-random substrate *Pad*

The mathematical model of McEliece HCC on modified elliptic codes (MEC) based on shortening (reduction of information symbols) or lengthening (adding information symbols) with damage is formally set by a set of input elements [2], given below.

Plain text M, consisting of information symbols *I*, at that $\forall \mathrm{I}_j \in GF(q)$:

– when shortening characters:

$$M_i = \{I_0, I_{h_1}, .. I_{h_j}, I_{k\text{-}1}\},$$

where h_j – information symbols equal to zero $|h| = 1/2k$, i.e. $I_i = 0,\ \forall I_i \in h$;

– when lengthening characters:

$$M_i = \{I_0, I_{h_{r_1}}, .. I_{h_{r_j}}, I_{k\text{-}1}\},$$

where h_r – extension information symbols *k.*

A plurality of closed texts (codograms):

$$C = \{C_1, C_2, \ldots, C_{q^k}\},$$

where $\forall A^*_{X_j} \in GF(q)$;

– when shortening characters:

$$C_i = (A^*_{X_0}, A^*_{h_1}, \ldots, A^*_{h_j}, A^*_{X_{n-1}});$$

– when lengthening characters:

$$C_i = (A^*_{X_0}, A^*_{h_{r1}}, \ldots, A^*_{h_{r_j}}, A^*_{X_{n-1}}).$$

A plurality of direct mappings (based on the use of public key – generating matrix):

$$\varphi = \{\varphi_1, \varphi_2, \ldots, \varphi_s\};$$

– when shortening characters:

$$\varphi_i : M \to C_{k-h_j},\ i = 1,2,\ldots,s;$$

– when lengthening characters:

$$\varphi_i : M \to C_{k-h_j}, i = 1,2,\ldots,s.$$

A plurality of reverse mappings (based on the use of a private (private) key – masking matrices):

$$\varphi^{-1} = \{\varphi_1^{-1}, \varphi_2^{-1}, \ldots, \varphi_s^{-1}\};$$

– when shortening characters

$$\varphi_i^{-1} : C_{k-h_j} \to M, i = 1,2,\ldots,s;$$

– when lengthening characters

$$\varphi_i^{-1} : C_{k-h_j} \to M, i = 1,2,\ldots,s;$$

A plurality of keys, parametrizing direct mapping (the public key of the authorized user):

$$K_{a_i} = \{K_{1_{a_i}}, K_{2_{a_i}}, \ldots, K_{s_{a_i}}\} = \{G_{X\ a_i}^{EC_1}, G_{X\ a_i}^{EC_2}, \ldots, G_{X\ a_i}^{ECs}\},$$

where $G_{X\ a_i}^{ECi}$ – generating $n \times k$ matrix of an algebrogeometric block disguised as a random *(n, k, d)*-code with elements from *GF*(*q*); a_i – ehe set of coefficients of a curve polynomial $a_1 \ldots a_6$, $\forall a_i \in GF(q)$, specifying a specific set of points on a curve from space P^2;

– when shortening characters:

$$\varphi_i : M \xrightarrow{K_{ia_i}} C_{k-h_j}, i = 1,2,\ldots,s;$$

– when lengthening characters:

$$\varphi_i : M \xrightarrow{K_{ia_i}} C_{k-h_j}, i = 1,2,\ldots,s.$$

A plurality of keys that parameterize the inverse mappings (private (private) key of the authorized user):

$$K^* = \{K_1^*, K_2^*, \ldots, K_s^*\} = \{\{X,P,D\}_1, \{X,P,D\}_2, \ldots, \{X,P,D\}_s\}, \{X,P,D\}_i = \{X^i, P^i, D^i\},$$

where X^i – masking non-degenerate randomly equiprobably generated by the key source $k \times k$ matrix with elements from *GF*(*q*); P^i – permutation randomly equally likely generated by the key

source $n \times n$ matrix with elements from $GF(q)$; D^i – diagonal generated by key source $n \times n$ matrix with elements from $GF(q)$, i.e. $\varphi_i : M \xrightarrow{K_{ia_i}} C_{k-h_j}, i = 1,2,\ldots,s$.

A plurality of defective texts CFT [2]:

$$CFT = \{CFT_1, CFT_2, \ldots, CFT_{q^k}\}.$$

A plurality of damage *CHD* [2]:

$$CHD = \{CHD_1, CHD_2, \ldots, CHD_{q^k}\}.$$

A plurality of direct damage (through the use of key – K^i_{MV2}, and algorithm *MV*2) [2]:

$$E = \{E^1_{K_{MV2}}, E^2_{K_{MV2}}, \ldots, \phi^S_{K_{MV2}}\}, i = 1,2,\ldots,s;$$

$f(x)_i$ – flag (damage, *CHD*); $C(x)_i$ – remainder (defective text, CFT); $f(x)=n-|C(x)|$, if $|C(x)|>y$, where y – some parametery, $y \in_Y Z_{q^m}, 0 < y < n$.

A plurality of mappings *MV2* F^r_n :

– is given by a bijective mapping between the set of permutations $\{S_1, S_2, \ldots, S_{2^n}\}$ and plurality $\# F^r_n, \quad \# F^r_n = \# \{(c,f)\} = 2^n!$.

A plurality of meaningful text (based on the use of the keya – K^i_{MV2}, and algorithm *MV*2):

$$E^{-1} = \{E^1_{K_{MV2}}{}^{-1}, E^2_{K_{MV2}}{}^{-1}, \ldots, E^S_{K_{MV2}}{}^{-1}\},$$

where $E^{-1}_{K_{MV2}} : \|f(x)_i\| + \|C(x)_i\| \to M, i = 1,2,\ldots,s;$ $f(x)_i$ — flag (damage, *CHD*), $C(x)_i$ – remainder (defective text, CFT);

$$f(x)=n-|C(x)|, \text{ if } |C(x)|> y,$$

where y – some parameter, $y \in_Y Z_{q^m}$. A plurality of key conversion codes flawed:

$$K^i_{MV2} \in K_{MV2}.$$

Algebrogeometric block (n, k, d)-code C_{k-h_j} (shortened) / C_{h_r} (lengthened) above $GF(q)$, i.e. a set of code words $C_i \in C_{k-h_j}$ (when shortening) / $C_i \in C_{h_r}$ (when lengthening), that the condition is satisfied $C_i \times H^T = 0$, where H – parity check matrix of an algebraic geometric block code; a_i – set defines a specific set of curve points from space P^2 to form the generating matrix; h_j – information symbols equal to zero, $|h|=1/2k$, i.e. $I_i=0$, $\forall I_i \in h$; h_r – information lengthening symbols k, $|h|=1/2k$, i.e. $I_i=0$, $\forall I_i \in h$.

Masking matrix mappings, given by a set of matrices $\{X, P, D\}_i$, where X – non-degenerate $k\times k$ matrix over $GF(q)$; P – permutation $GF(q)$ matrix over $GF(q)$ with one nonzero element in each row and in each column of the matrix; D – diagonal $n\times n$ matrix over $GF(q)$ with nonzero elements on the main diagonal; y – some parameter $y \in_Y Z_{q^m}, Z_{q^m} = \{0,1,\ldots 2^n - 1\}$; n – some parameter $n \in_Y Z_{q^n}, Z_{q^n} = \{1,\ldots 2^n\}$; a plurality of mappings $MV2\ F_n^r$.

Fig. 2.27 describes formation of a pseudo-random substrate based on McEliece hybrid crypto-code construction on flawed codes.

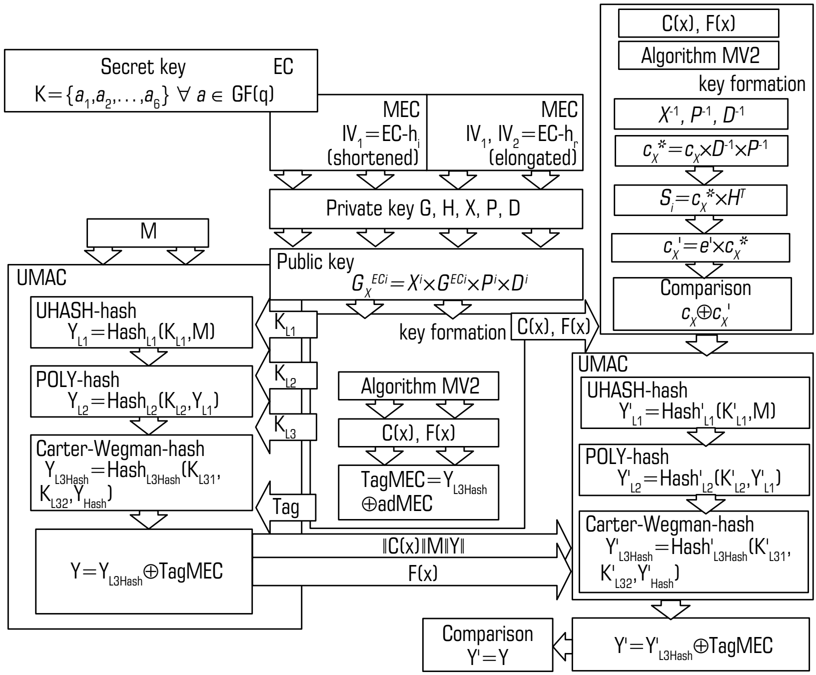


Fig. 2.27 Formation of a pseudo-random substrate based on McEliece hybrid crypto-code construction on flawed codes

Software implementation of the algorithm for the occurrence of collisional properties of hash codes

Using the reduced UMAC model (mini-UMAC), let's conduct research of the collision properties of message authentication codes, which consists in an experimental assessment of the distribution

of the number of collisions of the generated images. Reduced models are designed to investigate the main indicators of the efficiency of the cryptoalgorithm while maintaining its algebraic structure.

Since in the above UMAC scheme on the first layer (when generating a hash code) multiplicity of universal hashing functions are used we will carry out statistical research only on the second layer when forming a pseudo-random pad and at the final stage of generating authentication codes (after completing the summation).

When conducting statistical research of the collisional properties of the generated values of hash codes for each experiment, the mathematical expectations $m(n_1)$, $m(n_2)$ and $m(n_3)$, variances $D(n_1)$, $D(n_2)$ and $D(n_3)$ were estimated, as well as for a confidence level $P_{confid}(|\widetilde{m}(n_i)-m(n_i)|<\varepsilon)=0.98$ were determined based on the calculated accuracies $\varepsilon_1=t_{kp}(q(n_1)/\sqrt{n})$, $\varepsilon_2=t_{kp}(q(n_2)/\sqrt{n})$ and $\varepsilon_3=t_{kp}(q(n_3)/\sqrt{n})$ the corresponding extrema (lower and upper values) of the confidence intervals $(\widetilde{m}(n_i)-\varepsilon;\widetilde{m}(n_i)+\varepsilon)$. And besides $\widetilde{m}(n_i)$ it is a natural estimate for the mathematical expectation $m(n_i)$ of a random variable n_i, and $\widetilde{D}(n_i)$ is an estimate of the variance of the random variable n_i.

The research was carried out on a sample of size $N=10\,000$ elements. To form each element of the sample, the maximum was calculated over a set of $M=1\,000$ tuples of elements. Thus, the total volume of the generated sets was $N{\cdot}M=10^7$.

The obtained results of experimental studies are summarized in **Table 2.13**.

Let's analyze the obtained results of statistical research of the collision properties of message authentication codes: let's compare the obtained results of the average estimates of the mathematical expectations $m(n_1)$, $m(n_2)$ and $m(n_3)$ of the number of hashing rules with theoretical estimates (with the number $P_{coll}\cdot|H|$ – for the first criterion, with the number $|H|/|B|$ – for the second criterion and the number $P_{coll}{\cdot}H$ – for the third criterion), in this case.

Let's consider the first criterion by which the number of hashing rules is estimated for which there is a collision (coincidence of authentication codes) for two arbitrary input sequences. In accordance with theoretical estimates, this quantity is bounded from above by the number $P_{coll}\cdot|H|$. Let's concretize this (theoretical) estimate for authentication codes generated using the algorithms MASH-1, MASH-2, mini-UMAC MASH-1, mini-UMAC MASH-2, mini-UMAC AES, and mini-UMAC CCC.

The power of the key set for the algorithms MASH-1, MASH-2, mini-UMAC MASH-1, mini-UMAC MASH-2, mini-UMAC AES and mini-UMAC CCC is $|H|=2^{16}$, the cardinality of the set of generated authentication codes is also $|B|=2^{16}$. If to use the upper bound for the probability of collisions as the reciprocal of the power of the generated authentication codes $P_{coll}=2^{-16}$, let's get $n_1(x_1,x_2)\le P_{coll}\cdot|H|=1$ [5].

The collisional properties of the MASH-1 and MASH-2 encryption algorithms are significantly inferior to this upper theoretical estimate. In fact, the number of collisions on them is more than 7 times higher than the theoretical limit. Codes generated by all other algorithms also do not meet the first universality criterion, since the number of collisions exceeds the specified limit. Consequently, the criterion of universality is not satisfied by any of the algorithms.

Table 2.14 Results of experimental researches of collisional properties of authentication codes generated using MASH1, MASH2, mini-UMAC MASH1, mini-UMAC MASH2, mini-UMAC AES and mini-UMAC CCC (at P_{confid}=0,98)

Statistical characteristics of the experiment	MASH-1	MASH-2	mini-UMAC MASH-1	mini-UMAC MASH-2	mini-UMAC AES	mini-UMAC CCC
$\widetilde{m}(n_1)$	7.09*	7.14*	1.965	1.968	1.096	1.166
$\widetilde{D}(n_1)$	1.69	1.56	0.123	0.120	0.094	0.599
$\widetilde{m}(n_1)-\varepsilon_1$	7.061	7.111	1.957	1.961	1.088	1.148
$\widetilde{m}(n_1)+\varepsilon_1$	7.122	7.169	1.973	1.977	1.103	1.184
$\widetilde{m}(n_2)$	1.013	1.014	2.629*	2.64*	1.532	1.161
$\widetilde{D}(n_2)$	0.013	0.014	0.349	0.355	0.36	0.67
$\widetilde{m}(n_2)-\varepsilon_2$	1.01	1.011	2.62	2.63	1.52	1.14
$\widetilde{m}(n_2)+\varepsilon_2$	1.02	1.017	2.64	2.65	1.55	1.18
$\widetilde{m}(n_3)$	1.0008	1.0002	0.237**	0.224**	0.0005**	0**
$\widetilde{D}(n_3)$	$9993 \cdot 10^{-8}$	$9999 \cdot 10^{-8}$	0.184	0.177	$499 \cdot 10^{-6}$	0
$\widetilde{m}(n_3)-\varepsilon_3$	1.00006	0.9994	0.227	0.214	-2.087	0
$\widetilde{m}(n_3)+\varepsilon_3$	1.002	1.0009	0.247	0.234	0.001	0

** – natural estimates of mathematical expectations, according to which the number of collision values significantly exceeds their theoretical estimates;*
*** – natural estimates of mathematical expectations, according to which the number of collision values does not exceed their theoretical estimates*

Let's consider the second criterion by which the number of hashing rules is estimated under which the value of the authentication code does not change for an arbitrary input sequence. In accordance with theoretical estimates, this value for authentication codes generated using all algorithms is bounded from above by the number $|H|/|B|=1$.

The experimental results obtained indicate that the collision properties of authentication codes generated using the mini-MASH-1 and mini-MASH-2 algorithms do not satisfy the second criterion, since the number of collisions for them exceeds the theoretical limit by almost 3 times, and for all the rest algorithms also observe an excess of the permissible number of collisions. Consequently, the first criterion of strict universality is not satisfied by any of the algorithms.

In accordance with the third criterion, the number of hashing rules is estimated under which the corresponding values of the authentication code do not change for two arbitrary input sequences. The theoretical estimate of this value for universal hashing is bounded from above by the number $P_{coll} \cdot |H|$, which when using the upper bound for the collision probability $P_{coll} = 2^{-16}$ let's get $n_3 (x_1, x_2, y_1, y_2) \leq P_{coll} \cdot |H| = 1$ [5].

Thus, the presented approach provides the required indicators of stability, efficiency and universality in the post-quantum period, which makes it possible to build new complex mechanisms based on the synthesis of crypto-code constructions with hashing algorithms.

3 MODEL OF PROBABLE THREATS AND INFORMATION PROTECTION IN PUBLIC NETWORKS

ABSTRACT

The models of probable threats and protection of information in public networks reproposed. The most general model of the formal description of the protection system is the model of the security system with full overlap, in which a complete list of protection objects and threats to information is determined, and means of ensuring security are determined from the point of view of their effectiveness and contribution to ensuring the security of the entire telecommunications system. The described model also satisfies the general scheme of interactions, which includes the organizational management system, external and internal threats, as well as the environment of interaction between them. It follows from the comparison of the model with a complete overlap and the general scheme of interaction that the model does not take into account the possibility of simultaneous implementation of different types of threats, the power of influence, the possibilities of their interaction and, accordingly, preventive protective measures. Therefore, the development of the model of the formal description of the protection system is connected, first of all, with the consideration of a detailed description of the object of protection with the introduction of a more general concept of threats, as well as the construction of an apparatus for modeling various types of threats and their interactions.

It is also shown that the combination of four models (M1 – passive channel of information leakage, M2 – active channel of information leakage, M3 – unintentional unauthorized access, M4 – unauthorized access to information with the purpose of removing it) in different versions provides wide opportunities for modeling various known types of threats and their implementation. However, in connection with the continuity of the process of developing new and improving existing methods and means of implementing threats, it is necessary to use such approaches to ensuring information protection that allow detecting and preventing threats of unknown types and carrying out dynamic correction of protection behavior, adapting it to specific application conditions.

The basic M5 model is also described – unauthorized access to information for the purpose of removing it with an unknown type of threat, which is based on the M4 model, but a special entry channel for unknown types of threats is introduced into its structure, which is subject to identification and an adaptation loop. Based on the received diagnostic data, the transmission system adjusts the protection parameters (encryption, noise reduction, etc.). Thus, there is a continuous process of self-diagnosis with the subsequent change in the characteristics of the operating system without stopping it. Based on this, the information transmission system will turn into a feedback system, where the transmission channel affects the response of the system.

The M5 information protection model provides an opportunity for constant refinement of threat classes and response measures and continuous training of the adaptive component of the

information security tools (IST). Thus, the IST, built on the basis of this model, detects and prevents threats of unknown types.

In this regard, the basic model M6 is introduced – unauthorized access to information for the purpose of its removal with an unknown type of threat and an unspecified level of protection, which is based on the model M5 in order to obtain higher security. Its structure includes: a special module of internal diagnostics that diagnoses the entire protection system, makes a decision to adjust the behavior algorithm of the SHI, which allows to achieve fault tolerance of the IST; a special module that diagnoses the communication channel with subsequent changes in the security level, which allows to achieve the adaptability of the IST.

KEYWORDS

Information security, public networks, threat models, protection object, information security models.

Object of information protection

The most general model of the protection system formal description is the model of the security system with full overlap, in which a complete list of protection objects and threats to information is determined, means of ensuring security are determined from the point of view of their effectiveness and contribution to ensuring the security of the entire telecommunications system [62–64].

For this purpose, a set of protected objects $O=\{O_j\}$ is introduced into the model as a set of threats $T=\{T_j\}$, each of which is aimed at one or more protected objects. The set of threat-object relations creates a bipartite graph in which the edge $<t, O>$ exists if and only if the threat i is a mean of gaining access to the object O. It should be noted that one threat can target several objects, and one object can be affected by several threats.

The goal of the simulated security system is to cover all possible edges in the graph $\{<t, O>\}$, that is, to ensure that there is not a single unobstructed path to any object from any threat. This is achieved by introducing a third set $M=\{M_k\}$ of security means. In an ideal system, each mean $M_k \in M$ must remove at least one edge $<t, O>$ from graph $\{<T, O>\}$. Introducing the set M of security measures will transform a bipartite graph $\{<T, O>\}$ in a three partite $\{<T, M, O>\}$, containing arcs of the form $<t, m>$ and $<m, O>$.

Thus, in a protected system, any edge of the form $<t, O>$ defines an unprotected object. Here, the same security tool can cover more than one threat and / or protect more than one object. The described model also satisfies the general scheme of interactions, which includes the organizational management system, external and internal threats, as well as the environment of interaction between them.

It follows from the comparison of the model with a complete overlap and the general scheme of interaction that the model does not take into account the possibility of simultaneous

implementation of different types of threats, the power of influence, the possibilities of their interaction and, accordingly, preventive protective measures.

Therefore, the development of the formal description model of the protection system is connected, first of all, with the consideration of a detailed description of the object of protection with the introduction of a more general concept of threats, as well as the construction of an apparatus for modeling various types of threats and their interactions [65].

To describe the objects of protection, let's introduce a set $O=\{o(t)\}$, which describes the composition of objects that are protected. Taking into account the flowchart of the description of protection systems, let's additionally consider a set $C=\{c(r)\}$ or $C=\{c(i,j)\}$, which describes the connections (interactions) between the elements of the object, where $c(i,j)=1$, if there is a connection between the elements $o(i)$ and $o(j)$, and $c(i,j)$, if there is no connection.

It is obvious that this description of the object is in the form of an arbitrary network $<O, C>$ is the most general, however, in order to take into account the features of the considered model and solve problems related to obtaining estimates, it is necessary, first of all, for each element and connection to have appropriate characteristics (respectively, sets $H(O)$ and $H(C)$) and, secondly, for protection modeling, it is fundamentally important to distinguish qualitatively different connections (informational, control, etc.). Specifying the type of connection can be done in different ways, for example, in the $h(c)$ characteristics of the connection from the set $H(C)$, but since the sets of characteristics will be used only when obtaining estimates, for the simplicity of the formulation of the model we will introduce a list of relations $C=\left[C_1,C_2,...,C_n\right]$, where $C_1,C_2,...,C_n$ correspond to connections of the 1st, 2nd and other types.

Also, when describing the object of protection, it is necessary to take into account such a parameter as integrity, that is, the degree of damage and the level of operational efficiency of the object $P\left(<O,C>\right)$. Its introduction into the model of the object is expedient due to the use of testing of the protection object and internal diagnostics of the IPS. Thus, each subject $o(t)$ as part of the protection object O and the connection $c(i,j)$ will get efficiency ratio $P=p(c(i,j)=0...1$ and $P=p(o(t)=0...1$ respectively, where the coefficient p will vary in the interval from 0 (complete non-functionality – destruction) to 1 (full operability).

In addition, a complete description of the object is impossible without taking into account its external connections and the list of external objects interacting with it, which are marked: *EO* – for external objects and *EC* for connections with external objects. Since taking into account external interactions is especially important when building comprehensive assessments, the relevant sets of characteristics *H(EO)* and *H(EC)* are essential [5].

A model of a potential offender

In terms of the solved task of assessing the security of public networks, the description of the external environment should contain not only a description of external objects, but also a description of the alleged offender. In the simplest case, the offender is described by a set of external influences (threats) $T=\left\{t_i\right\}$ with the corresponding characteristics $H(T)=\left\{h(t)\right\}$. In the general case, it is necessary to consider different types of external influences: $T_1,T_2,...,T_n$.

This corresponds to the various goals of the offender: information capture, penetration, destructive actions, etc. Thus, in the general case, there is a list of sets: $T=\{t_i\}$ at $i=1,\ldots,N$, which is described by a corresponding list of characteristics $H(T)$.

In addition, the model must also take into account the so-called internal "semi-interactions", which correspond, on the one hand, to the possible impact on the object (an element of the object), and on the other hand, to the ability of the element itself to perform actions that are not foreseen by the technology and can have unintended consequences.

Let different possible effects on the elements and connections of the object of protection (vulnerability) be characterized by sets $U(O)$ and $U(C)$, where $U(O)=\{u(i)\}$ and $U(C)=\{u(i,j)\}$.

Similarly, the ability of an element (connection) to detect activity, i.e. to detect some impact unforeseen by information processing technology (for example, failure of an element (connection)), will be indicated $v(i)$ or $v(i,j)$ with sets $V(O)$ and $V(C)$ respectively, and sets $V(O)=\{v(i)\}$ and $V(C)=\{v(i,j)\}$, in turn, can be combined into a list V, as described above.

As components of the object description U and V must have the appropriate sets of characteristics $H(U)$ and $H(V)$ to receive estimates.

Threats are supposed to create pairs with different "vulnerabilities" – u from sets $U(O)$ and $U(C)$, that is, the "external influence" (threat from the violator) must correspond to the "possibility of such influence" (vulnerability) to create a pair (if such an external influence is not specified, in addition, for a specific object, such as $t=t(i)$ or $t=t(i,j)$). As a result of such an external link, the threat can be "built" both on the relevant technological (sanctioned) connections of the model, and on non-technological ones, which in general are pairs of the type: (v,u).

Therefore, the model describing the offender includes, in addition to immediate threats, the above-mentioned set of vulnerabilities U and internal influences V as factors contributing to the attack and as the only sources of influence in the absence of an external intruder.

Information protection means

Both natural restrictions (built-in protective properties) and artificial (additionally located equipment) – means of protection can prevent the spread of threats. Thus, the general formal description of the protection system should contain a list of protection means $Z=\{z\}$ with respective characteristics $H=\{h(z)\}$.

At the same time, it is important that each means of protection z must be associated with the corresponding element (elements) of the object to be protected or the link (links) between the elements. So, $z=z(i)$ or $z=z(i,j)$ (or similarly for groups of objects and connections $z=z(\{i\})$ and $z=z(\{i,j\})$). In addition, let's note that the means of protection $\{z\}$ are also elements of the object, i.e. can be subject to external and internal influences and, in turn, can themselves (as system elements) influence other elements [66, 67].

The development of the modeling apparatus with regard to the complication of models and the construction of methods for their combination is considered below [68].

To build more complex protection models consisting of basic models, it is advisable to use the tool of logical operations on simple information protection models.

Basic models of information protection

Models M1, M2, M3, M4 developed earlier in the works of V. Khoroshko [69].

Model M1. Passive information leakage channel

The model of passive information leakage assumes that the violator does not take active actions. Therefore, a set $T = \{t\}$ is sufficient to describe the violator, which describes such threats. In terms of the general model, such a set of threats can be denoted as T_1.

To describe the object, it is necessary to take into account:

O — individual elements of the object;

C – information connections (let them be labeled C_1 in terms of a general model);

U – uncontrolled elements, that is, those that can serve as sources of information for threats T_1 (notation U_1 can be entered in the general model).

In the description of the object, it is assumed that all elements are reliable and, accordingly, all other sets for the general model are empty.

Means of protection $Z = \{z\}$ for simplification, are assumed to be connected only to the elements of the object model, i.e. $z = z(o)$.

In this case, let's use the simplest predicates to describe the M1 model:

Object:

element(o) – if "*o*" belongs to the set *O*;

inf_conn(o, e) – if "o" and "e" belong to the set *O, and c(o, e)* to the set C_1;

not_contr(o) – if of element "*o*" it is possible to obtain information (that is, there is no guarantee that it is not possible to obtain information from this element), which corresponds to the set U_1 of general model.

Protection:

protect(o) – if there are protection means *z*, connected to "*o*";

(protect(o)) – if there are no protection means *(z)*, connected to "*o*".

Offender:

read(o) – if the offender tries to obtain information from "*o*" (of a specific element or the corresponding class, the type of elements from which an attempt is made to obtain information).

From this it follows that the M1 model represents a set of dangers $M_1 = \{m\}$.

Using the form of recording predicates adopted in the logical programming language PROLOG [69, 70] to describe the transfer of information through connections, the rules of derivation F_1, can be presented in the recursive form of the description of predicates:

m(o,t) ← *[(element(o)) & (not_contr(o/t)) & (protect(o)) & (read(o))];*

m(o/z,t) ← *((element(o)) & [m(e,t)&(inf_conn(o,e/c)) & (protect(o))]*∪*[m(e,t) & c(o,e) & (protect(o,z))].*

Predicates *(protect(o))* and *protect(o)* allow to build regardless of the presence of protection *z(o)* of information connections chain $c_i(o,e)$ of a view (**Fig. 3.1**).

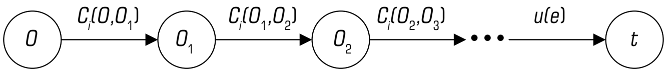


Fig. 3.1 The scheme of threat implementation in the M1 model

Moreover, among these chains there cannot be two identical ones (respectively, with and without protection means), which is ensured by the consistent use of predicates *(protect(o) and protect(o,z))*.

Since for each predicate "*m*" *describing "danger" are equated sets* {o}, {c}, {z}, {u}, {v}, as well as threats "*t*", for which this predicate is valid, then to determine the characteristics of the danger h(m) it is necessary to aggregate the initial evaluation data-characteristics of all components.

Model M2. An active leak channel

In the M2 model, it is assumed that the violator is engaged in collecting and removing information in all possible ways. Therefore, a set is sufficient to describe the violator $T = \{t\}$, describing such threats. In terms of the general model, such a set of threats can be denoted as T_1.

To describe the object, it is necessary to take into account:

O – individual elements of the object;

C – information connections (let them be denoted C_1 in terms of a general model);

U – uncontrolled elements, that is, those that can serve as sources of information for threats T_1 (notation U_1 can be entered in the general model);

V – internal influences, similar in many respects to T_1, aimed at establishing possible non-technological information connections *(v,u)*, and for simplicity (as noted in the general description of the object) can be combined into a separate set C_{nt} (or C_2).

So, in the description of the object, it is no longer assumed that all elements are reliable.

Means of protection $Z = \{z\}$ for simplification, are assumed to be connected only to the elements of the model, i.e. $z = z(o)$. In this case, to describe the M2 model, let's use the simplest predicates (in the extended form of the record).

Object:

element(o) – if "*o*" belongs to the set O;

inf_conn(o, e/c) – if "o" and "e" belong to the set O, *and c(o, e) to the set* C_1*;*

not_contr(o/u) – if of element "*o*" it is possible to obtain information (that is, there is no guarantee that it is not possible to obtain information from this element), which corresponds to the set U_1 of the general model;

ex_read(o/v) – if "*o*" can be activated to create a non-technological information channel purposefully or accidentally (V_1).

Protection:

protect(o/z) – if there are protection means z, connected with "*o*"*;*

protect(o) – if there are no protection means *(z)*, connected with "*o*".

Offender:

read(o/t) – if the offender tries to obtain information from "*o*" (of a specific element or the corresponding class, the type of elements from which an attempt is made to obtain information).

Then the M2 model will consist of the development of the description of the object through the introduction of the concept of "non-technological connection" in the form:

nt_conn(o,e/u,v) ← *(element(o))&(element(e))& ex_read(e/v)& not_contr(o/u)*

and the set of threats $M_2 = \{m\}$, and output rules F_2 will include rules:

m(o/u,t) ← *(element(o)) & (not_contr(o/u)) & (protect(o)) & (read(o/t));*
m(o/u,z,t) ← *(element(o)) & (not_contr(o/u)) & (protect(o/z)) & (read(o/t));*
$m(o/\{e\},\{c\},\{z\},\{u\},\{v\},t) \leftarrow (element(o))\&\{[m(e/u,t) \cup m(e/u,z,t) \cup m(e/\{a\},\{c\},\{z\},\{u\},\{v\},t) \;\&\; [(inf_conn(o,e/c)) \cup (nt_conn(o,e/u,v))] \;\&\; (protect(o))] \cup [(m(o/u,t) \cup m(e/u,z,t) \cup m(e/\{a\},\{c\},\{z\},\{u\},\{v\},t)) \;\&\; [(inf_conn(o,e/c)) \cup nt_conn(o,e/u,v))] \;\&\; (protect(o/z))]\}.$

Thus, the use of predicates:

nt_conn(o,e/u,v) ← *(element(o)) & (element(e)) & ex_read(e/v) & not_contr(o/u)* and *[(inf_conn(o,e/c)) ∪ nt_conn(o,e/u,v))] allows in the M2 model to form chains of information transmission consisting of both technological ones* $c_i(o,e)$, as well as from non-technological connections $c_{nt}(o,e)$ (possible informational influences) (**Fig. 3.2**).

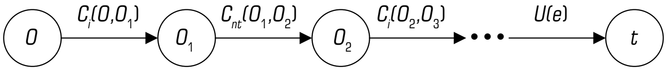


Fig. 3.2 Scheme of threats implementation in the M2 model

Moreover, among these chains (in fact, it is a set M_2) the same as in M_1, there can be no two identical chains. In the case of specific models, predicates of type "*nt_conn*" may depend on additional conditions, for example, on the presence of a suitable control connection:

nt_conn(o,e/u,v) ← *(element(o)) & (element(e)) & ex_read(e/v) & not_contr(o/u)ctrl_conn(o,e)*

if such a connection is provided in the model (as, for example, in the M3 model).

Model M3. Unintentional unauthorized access

In the M3 model, as well as in the model of passive information leakage, it is assumed that the offender shows only one type of active actions – unauthorized access (UAA) (penetration into the system). Therefore, a set $T = \{t\}$, describing such threats is sufficient to describe the offender. In terms of the general model, such a set of threats can be denoted as T_2 .

To describe the object, it is necessary to take into account:

O – individual elements of the object.

C – information connections (let them be denoted C_k or C_3 in terms of a general model).

U – uncontrolled elements, that is, those that can serve as the subject of UAA for threats T_2 (in the general model U_2).

V – internal influences, similar T_2 in many respects. They are aimed at establishing possible non-technological control relationships *(v,u)*, which for simplicity (as noted in the general description of the object) can be combined into a separate set C_{nt} (or C_4).

So, in the description of the object, as in the M2 model, it is no longer assumed that all elements are reliable.

Means of protection $Z = \{z\}$, just as in models M1 and M2, are assumed to be connected only to the elements of the object model, i.e. $z = z(o)$, and do not interact with other elements of the model and external objects, that is, flawless and reliable.

In this case, let's use the simplest predicates to describe the M3 model.

Object:

element(o) – if "*o*" belongs to the set *O*;

ctrl_conn(o, e/c) – if "o" and "e» belong to the set *O*, *and c(o, e)* belongs to the set C_k *;*

available(o/u) – if to element "*o*" possible access to change its mode of operation, which corresponds to the set U_2 of the general model;

ex_uaa(o/v) – if element "*o*" can be activated to create a non-technological control effect either purposefully or accidentally (V_2).

Protection:

ua_prot(o/z) – if there are protection means *z*, connected with "*o*"*;*

ua_prot (o) – if there are no protection means *(z)*, connected with "*o*"*.*

Offender:

uaa(o/t) – if the offender tries to gain access to "*o*" (of a specific element or an element belonging to a class or type of elements to which UA is attempted).

Then the M3 model, similarly to M2, will consist of the development of the description of the object through the introduction of the concept of "non-technological control (unauthorized) communication" in the form:

$$nt_conn(o,e/u,v) \leftarrow (element(o))\ \&\ (element(e))\ \&\ ex_uaa(o/v)\ \&\ available(e/u)$$

and the set of threats $M_3 = \{m\}$, and output rules F_3 will include rules:

m(o/u,t) ← (element(o)) & (available(o/u)) & (uaa_prot(o)) & (uaa(o/t));
m(o/u,z,t) ← (element(o)) & (available(o)) & (uaa_prot(o/z)) & (uaa(o/t));
m(o/{e},{c},{z},{u},{v},t) ← (element(o)) & {[m(e/u,t) ∪ m(e/u,z,t) ∪ m(e/{a},{c},{z},{u},{v},t) & [(ctrl_conn(o,e/c)) ∪ (ua_conn(o,e/u,v))]& (uaa_prot(o))] ∪ [(m(o/u,t) ∪ m(e/u,z,t) ∪ m(e/{a},{c},{z},{u},{v},t))& [(ctrl_conn(o,e/c)) ∪ ua_conn(o,e/u,v))] & (uaa_prot(o/z))]}.

Thus, the use of predicates:

nt_conn(o,e/u,v) ← (element(o)) & (element(e)) & ex_uaa(o/v) & available(e/u) and *(ctrl_conn(o,e/c)) ∪ ua_conn(o,e / u,v)) allows in the M3 model to form chains of control influences consisting of both technological* connections $c_k(o,e)$, as well as from non-technological *ones* $c_{nk}(o,e)$ (possible access interactions) (**Fig. 3.3**).

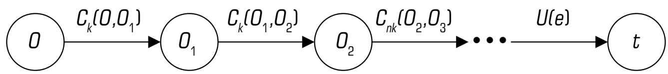


Fig. 3.3 Scheme of threats implementation in the M3 model

Moreover, among these chains (in fact, it is a set M_3) the same as in M_2, there can be no two identical chains.

Unlike model M2, in model M3 the direction of connections changes.

Model M4. Unauthorized access to information for the purpose of its obtaining

The M4 model combines the M2 and M3 models. The model "UAA to information with the aim of removing it" or "active leakage of information" assumes that the offender takes active actions – performs unauthorized access (penetration into the system) to influence its elements in order to activate them to obtain information (that is, new "given" internal influences for building new non-technological information transmission channels or changing existing characteristics). So, a list $T = [T_1, T_2]$ is enough to describe the violator, where T_1 – threats to withdraw information, and T_2 — UAA threats.

To describe the object, it is necessary to take into account:

O – individual elements of the object;

C – information connections (C_1);

C_k – control connections (C_3);

U_1 – uncontrolled elements, that is, those that can serve as sources of information for threats T_1;

U_2 – uncontrolled elements, that is, those that can serve as the subject of UAA for threats T_2;

V_1 – internal influences, similar in many respects to T_1, which are aimed at establishing possible technological information connections (v_1, u_1), and which can be combined into a separate set C_{nt} (or C_2).

V_2 – internal influences, similar in many respects to T_2. They are aimed at establishing possible non-technological control relationships (v_2, u_2), which can be combined into a separate set C_{nt} (or C_4).

So, in the description of the object, as in the M2 model, it is no longer assumed that all elements are reliable.

Means of protection $Z = \{z\}$, just as in models M2 and M3, are assumed to be connected only with the elements of the object model, i.e. $z = z(o)$, and do not interact with other model elements and external objects (are flawless and reliable).

In this case, let's use the simplest predicates to describe the M4 model.

Object:

element(o) – if "*o*" belongs to the set O;

inf_conn(o, e/c$_1$*) – if "o" and* "e» belong to the set O, *and c(o, e)* belongs to the set C_1;

ctrl_conn(o, e/c$_2$*) – if "o" and* "e» belong to the set O, *and c(o, e)* belongs to the set C_k;

not_contr(o/u$_1$*)* – if it is possible to obtain information from the "o" element (that is, there is no guarantee that it is not possible to remove information from this element), which corresponds to the set U_1 of the general model;

available(o/u$_2$*)* – if the "o" element can be accessed to change its mode of operation, corresponding to the set U_2 of the general model;

ex_read(o/v$_1$*)* – if the "o" element can be activated to create a non-technological information channel either purposefully or accidentally (V_1);

ex_uaa(o/v$_2$*)* – if the "o" element can be activated to create a non-technological control effect either purposefully or accidentally (V_2).

Protection:

protect(o/z$_1$*)* – if there are protection means associated with "o"*;*

protect(o)) – if there is no protection mean *(z)*, connected to "o"*.*

uaa_prot(o/z$_2$*)* – if there are UAA protection means associated with "o"*;*

uaa_prot(o) – if there are no means of protection against UAA*.*

Offender:

read(o/t$_1$*)* – if the offender tries to receive information from "o" (a specific element or a corresponding class, type of elements from which an attempt is made to receive information);

uaa(o/t$_2$*)* – if the offender attempts to access "o" (a specific element or an element belonging to a class, type of elements, to which the UAA is attempted).

Then the M4 model will consist of a special development of the description of the object through the introduction of new concepts:

a) "non-technological control (unauthorized) communication":

na_conn(o,e/u$_2$*,v*$_2$*)* $\leftarrow$ *(element(o)) & (element(e)) & ex_uaa(o/v*$_2$*) & available(e/u*$_2$*)*;

b) "unauthorized controlling influence":

$influence(o/u_2,t_2) \leftarrow (element(o))\ \&\ (available(o/u_2))\ \&\ (uaa_prot(o))\&(uaa(o/t_2));$
$influence(o/u_2,z_2,t_2) \leftarrow (element(o))\ \&\ (available(o/u_2))\ \&\ (uaa_prot(o/z_2))\ \&\ (uaa(o/t_2));$
$influence(o / \{e_2\},\{c_2\},\{z_2\},\{u_2\},\{v_2\},t_2) \leftarrow (element(o))\ \&\ \{[influence(e / u_2,t_2)) \cup$
$influence(e / u_2,z_2,t_2) \cup influence(e / \{a\},\{c_2\},\{z_2\},\{u_2\},\{v_2\},t_2)\ \&$
$[(ctrl_conn(e,o / c_2)) \cup ua_conn(e,o / c_2,v_2))]\ \&\ (uaa_prot(o))]\} \cup$
$influence(o / u_2,t_2) \cup influence(o / u_2,z_2,t_2) \cup influence(e / \{a\},\{c\},\{z_2\},\{u_2\},\{v_2\},t_2)\ \&$
$[(ctrl_conn(e,o / o,c_2)) \cup ua_conn(e,o / u_2,v_2))]\ \&\ (uaa_prot(o,z_2))]\};$

c) "non-technological information communication":

$nt_conn(o,e / u_1,v_1,\{a\},\{c_2\},\{z_2\},\{u_2\},\{v_2\},t_2) \leftarrow element(o)\&\ element(e)\&$
$[not_contr(o / u_1)\ \&\ ex_read(e/v)] \cup influence(e / \{a\},\{c_2\},\{z_2\},\{u_2\},\{v_2\},t_2).$

Then the set of threats $M_4 = \{t\}$ given by the output rules F_4.

$m(o / u_1,t_1) \leftarrow (element(o))\ \&\ (not_contr(o / u_1)\ \&\ (protect(o))\ \&\ (read(o / t_1));$
$m(o / u_1,z_1,t_1) \leftarrow (element(o))\ \&\ (not_contr(o / u_1))\ \&\ (protect(o / z_1))\ \&\ (read(o / t_1));$
$m(o / \{e\},\{c_1\},\{z_1\},\{u_1\},\{v_1\},\{c_2\},\{z_2\},\{u_2\},\{v_2\},t_1,t_2) \leftarrow (element(o))\ \&\{[m(e / u_1,t_1) \cup$
$m(e / u_1,t_1) \cup m(o / \{e\},\{c_1\},\{z_1\},\{u_1\},\{v_1\},\{c_2\},\{z_2\},\{u_2\},\{v_2\},t_1,t_2)\ \&\ (protect(o))] \cup$
$[(m(o / u,t) \cup m(o / u,z,t).$

The structure of connections described by the M4 model is shown in **Fig. 3.4** and **3.5**.

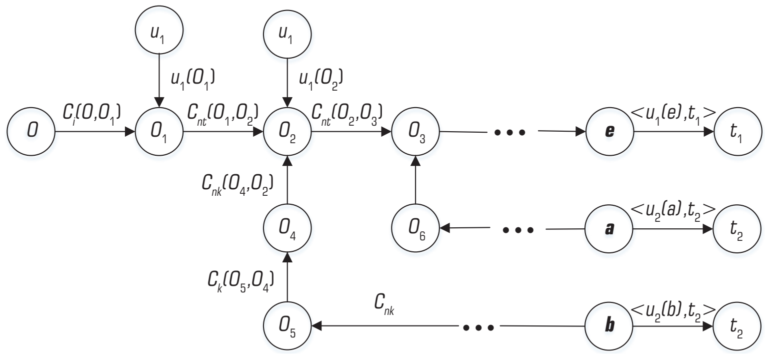


Fig. 3.4 Scheme of implementation of threats in the M4 model, taking into account non-technological control connections

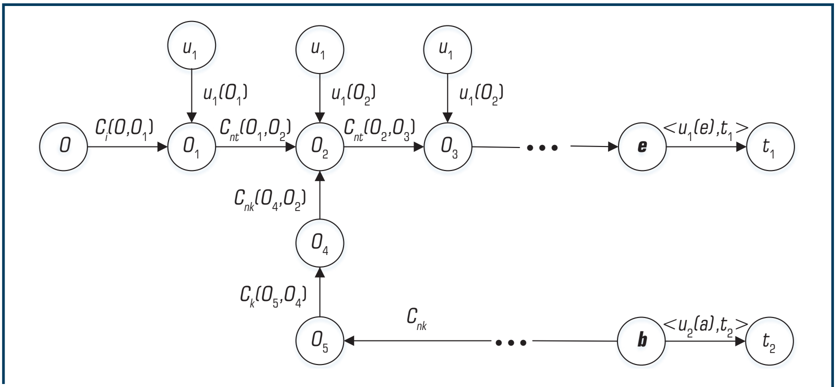


Fig. 3.5 The structure of threats in the M4 model taking into account unauthorized control connections

The given examples illustrate, on the one hand, the capabilities of the proposed apparatus for the formal description of various models and the clarity of their presentation, and on the other hand, the complexity of modeling each specific protection system.

3.1 CONSTRUCTION OF BASIC INFORMATION PROTECTION MODELS BASED ON SIMPLE MODELS

The introduced concept of unification can be expanded to unify two models, which will allow building new, more complex models, starting from simple ones.

The combination of models can be depicted in the following way:

$$M = M_i \cup M_j,$$

if the simplest predicates of description are combined in ***M***, and the predicates of "realized threats" – threats are supplemented by a predicate of the type: $m \leftarrow m_i \cup m_j$. This will make it possible to distinguish threats of different models M_i , M_j and avoid errors in recursive definitions m_i, m_j .

For example, a model $M = M - t \cup M + t$ has a view:

element(o) – if "*o*" belongs to the set *O*;

inf_conn(o, e/c =) – if "o" and "e» belong to the set *O*, *and c(o, e)* belongs to the set *C–t;*

not_contr(o/u-) – if it is possible to obtain information from the element "o" (that is, there is no guarantee that it is not possible to remove information from this element), which corresponds to the set *U–t*;

read(o/t-) – if the offender tries to receive information from "o" (a specific element or a corresponding class, type of elements from which an attempt is made to receive information);

ctrl_conn(o, e/c+,) – if "o" *abd* "e" belong to the set *O*, *and c(o, e)* belongs to the set *CM+t;*

available(o/u+) – if the "o" element can be accessed to change its mode of operation, corresponding to the set *U+ t*;

ex_uaa(o/t+) – if the violator tries to get information from "about" (a specific element or a related class, the type of elements from which unauthorized access is attempted).

Output rules:

$m - t(o) \leftarrow [(element(o)) \ \& \ (not_contr(o)) \ \& \ (read(o)] \cup [m - t(e)\&(inf_conn(o,e))];$

$m + t(o) \leftarrow [(element(o)) \ \& \ (available(o)) \ \& \ (uaa(o)] \cup [m + t(e) \ \& \ (ctrl_conn(o,e))];$

$m(o) \leftarrow m - t(o) \cup m + t(o)$

or in complete form:

$m - t(o \,/\, \{a\},\{c\text{-}\},\{u\text{-}\},t\text{-}) \leftarrow [(element(o)) \ \& \ (not_contr(o/u\text{-})) \ \& \ (read(o/t\text{-})] \cup$

$[m - t(e \,/\, \{a\},\{c\text{-}\},\{u\text{-}\},t) \ \& \ (inf_conn(o,e \,/\, c\text{-}))];$

$m + t(o \,/\, \{e\},\{c+\},\{u+\},t+) \leftarrow [(element(o)) \ \& \ (available(o/u+)) \ \& \ (uaa(o/t+)] \cup$

$[m + t(e \,/\, \{a\},\{c+\},\{u+\},t+) \ \& \ (ctrl_conn(o,e/c+))];$

$m(\, o \,/\, \{e\},\{c_1\},\{u_1\},t_1,\{c_2\},\{u_2\},t_2\,) \leftarrow m - t(o) \cup m + t(o).$

By analogy with unification, the operation of composition (or product) of models is introduced: $M = M_i \cdot M_j$ in which the simplest predicates are combined, and the main components of interactions – communication, threats and vulnerability *M* are replaced by dangers, respectively M_j. Thus, instead of the simplest *t(o)*, *v(o)*, *u(o)* or *c(o,e)* chains of realized threats may appear: $m_i(o)$ or $m_i(o,e)$.

Therefore, for certainty, it is necessary to enter accordingly:

$t \cdot -t$ – *t-product* (*t(o)* replaced by *τ(o)*);

$v \cdot -v$ – *v-product* (*v(o)* replaced by *τ(o)*);

$u \cdot -u$ – u-*product (u(o)* replaced by *τ(o)*);

$c \cdot -c$ – c-*product* (c(o,e*)* replaced by *m*(o,e*)*).

For example, the product of models $M = M - tc \cdot M - v$ allows to build a species model $M(O,C,U,V,T) = M - t(O, M - v(O,C,F),T)$:

element(o) – if "o" belongs to the set *O*;

inf_conn(o, e/c =) – if "o» *and* "e" belong to the set O, *and c(o, e)* belong to the set *C;*

not_contr(o/u) – if it is possible to obtain information from the "o" element (that is, there is no guarantee that it is not possible to remove information from this element), which corresponds to the set *U*;

read(o/t) – if the offender tries to obtain information from "o" (a specific element or the corresponding class, type of elements from which an attempt is made to receive information);

ex_read(o / v) – if the "o" element can be activated to create a non-technological information channel either purposefully or accidentally (*V*).

Output rules:

$m - v(o,e) \leftarrow [(element(o)) \,\&\, (element(e) \,\&\, (ex_read(e) \,\&\, not_contr(o)] \cup$
$[m - v(o,i) \,\&\, (inf_conn(i,e)];$
$m - t(o) \leftarrow [(element(o)) \,\&\, not_contr(o) \,\&\, read(o)] \cup [m - t(e) \,\&\, m - v(o,e)];$
$m(o) = m - t(o / \{c\}) = m - t(o,e).$

This model considers all possible leakage channels only through non-technological channels that also use technological information links.

In order to fully reproduce the M2 model – "information leakage" described above in the examples, it is necessary to take into account the means of protection and "pure" technological channels of information transmission, which are not taken into account in the example with the product, since the predicate $m - v$ includes only a combination of technological and non-technological connections. In such a way:

$$M2 = (M - tc \cdot M - v) \cup M - t \cup Z.$$

Similarly for the M3 model there is:

$$M3 = (M + tc \cdot M + v) \cup M + t \cup Z.$$

To reproduce the M4 model, it is necessary to combine the following models:

$(M + t_2c_2 \cdot M + v_2)$ – unauthorized access through non-technological channels;

$(M + t_2c_2 \cdot M + v_2) \cup M + t_2$ – unauthorized access through non-technological and technological channels;

$(M - v_1v_1 \cdot ((M + t_2c_2 \cdot M + v_2) \cup M + t_2))$ – violation of information dissemination regime due to unauthorized access;

$((M - v_1v_1 \cdot ((M + t_2c_2 \cdot M + v_2) \cup M + t_2)) \cup M - v_1)$ – violation of information dissemination regime due to unauthorized access;

$(M - t_1c_1 \cdot ((M - v_1v_1 \cdot ((M + t_2c_2 \cdot M + v_2) \cup M + t_2)) \cup M - v_1)$ – leaks through non-technological channels arising as a result of violation of the information dissemination regime, including through unauthorized access.

As a result:

$$M4 = (M - t_1c_1 \cdot ((M - v_1v_1 \cdot ((M + t_2c_2 \cdot M + v_2) \cup M + t_2)) \cup M - v_1) \cup Z).$$

Thus, using the introduced operations of union and product of the simplest models, it is possible to obtain basic models of protection. The considered synthesis technique can be extended to perform the introduced operations with basic models.

3.2 MODELS OF ACTIVE DYNAMIC INFORMATION PROTECTION

Formal presentation of a comprehensive model of information protection in public networks.

It follows from the above that for a complete description of the composition of the protection model, in the general case, it is necessary to consider the following three sets:

1) $< O, C, EO, EC >$;
2) $< T, U, V >$;
3) $< Z >$.

In each specific case, these sets can be considered in a truncated form. The model of the protection system ***M*** consists of a set of "realized" threats – dangers $\{m\}$, which should be derived from the description of the composition of the protection system:

$$M \leftarrow F(O,C,U,V,EO,EC,T,Z),$$

where F – rules for output for a specific model, that is, for each model M_i, there must be their own output rules F_i, which allows to get a complete description of a specific protection system, taking into account the possible actions of the violator.

Comprehensive assessment for each model M_i is built by aggregating the characteristics of the components of the model, first, to obtain characteristics for each *m* from M_i, based on the initial characteristics *i*, secondly, to obtain generalized indicators for the model M_i in general.

It is quite obvious that, despite the huge variety of objects of protection, threats, means of protection, etc., when building a complex model of protection, some basic elements can be used, which allow to vary certain specific conditions.

Model M5. Unauthorized access to information for the purpose of obtaining it with an unknown type of threat

The combination of four models (M1 – passive channel of information leakage, M2 – active channel of information leakage, M3 – unintentional unauthorized access, M4 – unauthorized access to information with the aim of removing it) in various variants provides wide opportunities for modeling various known types of threats and their implementation [67–72]. However, in connection with the continuity of the process of developing new and improving existing methods and means of implementing threats, it is necessary to use such approaches to ensuring information protection that allow detecting and preventing threats of unknown types and carrying out dynamic correction of protection behavior, adapting it to specific application conditions.

In this connection, the basic model M5 is introduced – unauthorized access to information for the purpose of removing it with an unknown type of threat. The M5 model (**Fig. 3.6**) is based on the M4 model [68], but a special entry channel for unknown types of threats, which are subject to identification and adaptation contour, is introduced into its structure.

Here $\{O_i\}$ – objects of protection; $\{t_i\}$ – set of threats; $\{C_{ij}\}$ – set of information connections; $\{U_{ij}\}$ – set of vulnerabilities of objects and connections; $\{C_k\}$ – set of control connections;

a, b, e – unprotected objects; t_n – threats of unknown types; *AR* – contour of adaptive protection control; $C_{0:}$ – adaptive control communication.

At the same time, two types of uncertainties are considered:

– parametric uncertainty – parameters of the object and threats belong to a specific known field of definition, where they change according to unknown laws;

– signal uncertainty – object and threat models given by the function of time.

To implement the adaptation mechanism, a traditional scheme of a two-level adaptive control system is used, which includes the main circuit and the adaptation circuit. At the level of the main circuit, the reference mathematical model developed in this section and the control algorithm built on the basis of the reference model for known types of threats are specified.

With the help of the adaptation algorithm, the vector of adaptive control connections in the main circuit is adjusted to achieve the goal of protection against unknown types of threats and unknown initial parameters of the protection object.

Along with the probability of detecting unknown types of threats, one of the main indicators of the M5 model is the promptness of response. The assessment of responsiveness is based on the description of the transition process of the information protection system from the state N_j in N_{j+1}.

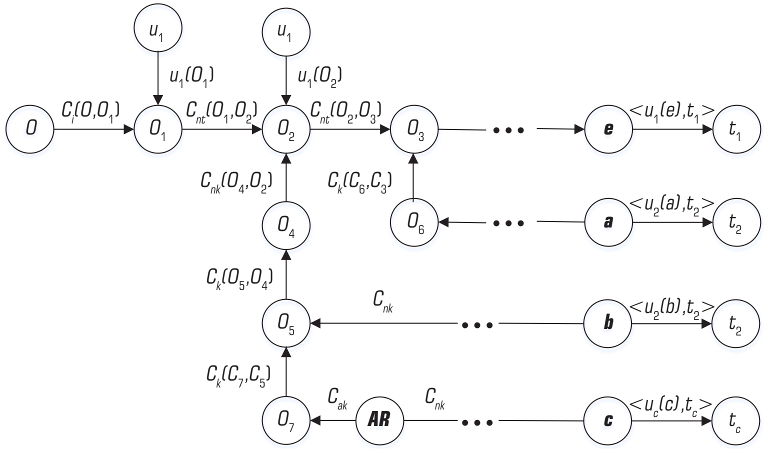


Fig. 3.6 The structure of threats in the M5 model, taking into account the adaptive protection contour

Let the initial moment of the period of functioning of the information protection system be the actual time of the start of the threat implementation, and the final moment – the completion of

operational measures. Let's mark $\Delta t_j = t_{i+1} + t_i$, – the total time of the change in the state of the IPS when the threat is realized. Then:

$$\Delta t_j = t_i^{\text{det.}} + t_i^{\text{clas.}} + t_i^{\text{tr.}} + t_i^{\text{res.}},$$

where $t_i^{\text{det.}}$ – threat detection time; $t_i^{\text{clas.}}$ — time of the threat classification (identification of the type); $t_i^{\text{tr.}}$ – time of transmission of information in IPS; $t_i^{\text{res.}}$ — time of response measures implementation.

At the same time, for a one-time impact of the threat, these temporary indicators will be determined in the following way:

$$t_i^{\text{det.}} = f_i(M,R,P,K),$$

$$t_i^{\text{clas.}} = f_2(M,R,P,K),$$

$$t_i^{\text{tr.}} = f_3(a,s),$$

$$t_i^{\text{res.}} = f_3(P,a,s),$$

where M – a number of threat detection methods; R – set of indicators characterizing methods of implementing the detection complex; P – set of threat indicators; K – set of threat classes; a – performance of technical means of the IPS; s – data transfer rate.

Therefore, the total time

$$\Delta t_j = F(M,R,P,a,s,K),$$

is not a constant value, and a condition must be introduced when operating the PS with calculations of the M5 model

$$\Delta t_j \le T_{i\,\text{lim}},$$

where $T_{i\,\text{lim}}$ – limitations on the promptness of response.

In order to fulfill this condition, it is necessary to constantly refine the classes of threats and response measures, which are carried out simultaneously with the continuous training of the adaptive component of the IPS.

For a complete description of the composition of the M5 model, in general, consider the following sets:

1. $< O, c, Eo, Ec, a >$ – object of protection;
2. $< T, U, V, a >$ – intruder;
3. $< z, t >$ – means of protection,

where *O* – objects to be protected, *c* – a set of information and control connections, *Eo, Ec* – external communication objects, *T* – set of threats, *U* – set of vulnerabilities of objects and connections, *V* – internal influences aimed at establishing non-technological control relationships, *z* – a set of protections, *t* – time, *a* – uncertainty parameter.

Model M6. A model of unauthorized access to information for the purpose of its removal with an unspecified level of protection and an unknown type of threat

The M5 information protection model presented in the previous subsection provides an opportunity for constant refinement of threat classes and response measures and continuous training of the adaptive component of the IPS. Thus, the IPS, built on the basis of this model, detects and prevents threats of unknown types.

However, due to the fact that the characteristics of the data transmission channel are constantly changing and the information protection system requires continuous control, it is necessary to use such approaches to ensure information protection that allow:

– to detect changes in the characteristics of the communication channel and carry out dynamic correction of the level of protection, adapting it to specific application conditions;

– to continuously diagnose communication systems with further adjustment of protection means.

In this regard, the basic model M6 is introduced – unauthorized access to information for the purpose of its removal with an unknown type of threat and an unspecified level of protection [67, 72].

The M6 model (**Fig. 3.7**) is based on the M5 model in order to obtain higher protection.

Its structure includes: a special module of internal diagnostics that diagnoses the entire protection system, makes a decision to adjust the behavior algorithm of the IPS, which allows to achieve fault tolerance of the IPS; a special module that diagnoses the communication channel with subsequent changes in the security level, which allows to achieve the adaptability of the IPS.

Notations of the M6 model: $\{O_i\}$ – objects of protection; $\{t_i\}$ – set of threats; $\{C_{ij}\}$ – a set of information connections; $\{U_{ij}\}$ – set of vulnerabilities of objects and connections; $\{C_k\}$ – set of control connections; *a, b, e* – unprotected objects; t_n – threats of unknown types; *AR* – contour of adaptive protection control; $C_{0:}$ – adaptive control communication; *ID* – internal diagnostics module; *CD* – communication channel diagnostics module; C_{ik} – diagnostic relationship between *ID* and IPS; C_{ck} – communication that provides information about the state of the communication channel for the IPS using the module *CD.*

Next, let's describe the changes made to the M5 model in order to obtain higher security in the M6 model.

Let's provide a detailed description of the functioning algorithm of the internal diagnostics module of the IPS (ID) of the M6 model [73–75].

Depending on changing external factors (i.e. whether the system is affected by an intruder, and if so, what) the system will adopt the following states:

1. E_0 – proper operation of the system *S*.
2. $E_1,\ldots,E_n$ – various types of malfunctions associated with the corresponding types of threats.

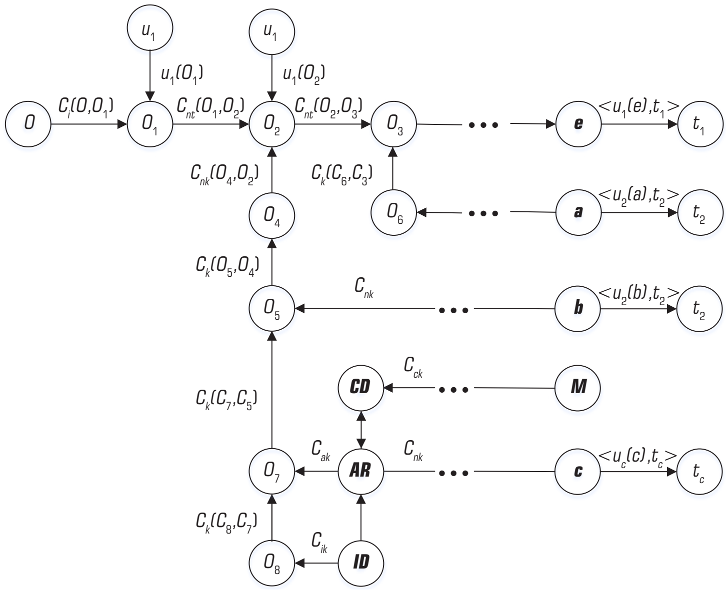


Fig. 3.7 The structure of the M6 model. Increased efficiency of protection is provided by components of internal diagnostics and diagnostics of the communication channel

Internal diagnostics is closely related to the identification of the transition operator H, which will transform the input signal $e(\cdot)$ into the output $J(\cdot) = H(e(\cdot))$ with the help of a model $\widehat{H}$ while minimizing the identification error $\varepsilon(\cdot) = J(\cdot) - \widehat{H}(e(\cdot))$.

Since the transition operator $H = f(x(\cdot,e))$ depends on the internal state of the studied system S, diagnosis is further carried out using the classification of identified internal states $x(\cdot,e)$, such that $\widehat{H} = f(\widehat{x}(\cdot,e))$, by means of structural distance d_x.

Identification actions will be performed using the model method using the second-order gradient algorithm.

The objective is to identify the system S (not necessarily linear), discretized with a period during the duration $T = N\tau$ using a linear model $\widehat{H}$, which depends on p parameters: $H = g(x), X = [x_1,\ldots,x_p]$; X ε, for example, by the vector of coefficients of the transfer function associated with $\widehat{H}$. In this case, let's assume that the output $J(\cdot)$ is one-dimensional [74].

Identification criterion I, which is to be minimized with respect to X, represents the root mean square error on the segment $[0,T]$, when $\widehat{e}(\cdot)$ is considered given:

$$I(X) = \sum_{l=o,N} \varepsilon^2(l\tau) = \sum_{l=0,N} J(l\tau) - \widehat{H}(e(l\tau)) = \sum_{l=0,N} J(l\tau) - \widehat{J}(l\tau). \quad (3.1)$$

The specified criterion (1), depending on the goal, can be replaced by any of the following alternative criteria.

Identification corresponds to the solution of a nonlinear optimization problem [75]:

$$I(\widehat{X}) = \min_X (I(X).$$

For example, using the gradient method, where the gradient $\nabla I(X)$ is a vector:

$$\nabla I(X) = \left[\frac{\partial I(X)}{\partial x_i}, i = 1, \dots p \right], \frac{\partial I(X)}{\partial x_i} = -2 \sum_{l=0,N} \varepsilon(l\tau) \frac{\partial \widehat{J}(l\tau)}{\partial x_i}$$

Values $\partial \widehat{J}(l\tau) / \partial x_i$ represent the sensitivity of the model $\widehat{H}$ to parameter changes *X.* The gradient method consists in making successive movements in the direction of $\nabla I(X^m)$ from the point X^m in order to minimize $\nabla I(X^m + \delta x)$ in the indicated direction by adjusting the step $\delta X = X^{m+1} - X^m$; in this case, the process resumes at the point X^{m+1} .

If to decompose $I(X^m + \delta x)$ to the second order and if to write that X^{m+1} is a stationary point, then:

$$I(X^m + \delta x) = I(X^m) + {}^t \nabla I(X^m) \delta x + \frac{1}{2} {}^t \delta X \nabla I(X^m) \delta X; \tag{3.3}$$

$$\frac{\partial I(X^m + \delta X)}{\partial x} = 0 \Rightarrow \delta x = -\left[\nabla^2 I(X^m) \right]^{-1} \nabla I(X^m); \tag{3.4}$$

$$X^{m+1} = X^m - k \left[\nabla^2 I(X^m) \right]^{-1} \nabla I(X^m), k \geq 0. \tag{3.5}$$

Value:

$$\nabla I(X) = \left[\frac{\partial^2 I(X)}{\partial x_i \partial x_j}, i = 1, \dots p; j = 1, \dots, p \right] = \sum_{i=0,N} \left(\nabla \widehat{J}(l\tau)^t \nabla \widehat{J}(l\tau) + \varepsilon(l\tau) \nabla^2 \widehat{J}(l\tau) \right)$$

is the sum of two types of terms, the first of which is related to the sensitivity of the model $\widehat{H}$, and the others are negligible near the optimum $\widehat{X}$ due to the factor $\varepsilon(l\tau)$. Thus, the members $\varepsilon(l\tau) \nabla^2 \widehat{j}(l\tau)$ can be neglected when calculating the above quantity with successive iterations *m.*

Like all identification procedures, this procedure requires data about the initial state X^0 of the system *S*. Most diagnosis tasks use good knowledge of system state E_0 which corresponds to the specified to add the state X^0 parametric value that corresponds to it. If necessary, it is possible

to carry out identification based on the model $\widehat{H}$ of a small order, in order to gradually improve it after successive identifications with increasing *p* values.

It is necessary to briefly indicate the modifications necessary to account for measurement noise. B_l – noise during the measurement $J(l\tau)$ output signal $J(\cdot)$ at a moment in time $(l\tau)$. The measured random variable in this case is defined as:

$$\delta^2(l\tau) = J(l\tau) + B_l, \overline{B}_l = 0, \tag{3.6}$$

and let $\delta\widehat{X}$ – the resulting random error for the parameters identified at the optimum *X* for the model $\widehat{H}$.

It is shown that:

a) average error value $\delta\widehat{X}$ is:

$$\delta\widehat{X} = B^{-1}\beta, B = \nabla I(X) = \sum_{i=0,N} \nabla_{\widehat{X}}\widehat{J}(l\tau)^t \nabla_{\widehat{X}}\widehat{J}(l\tau) = \beta \sum_{i=0,N} B_l \nabla_{\widehat{X}}\widehat{J}(l\tau); \tag{3.7}$$

b) covariance error matrix $\delta\widehat{X}$ is equal to V^2B^{-1}, where:

$$V = \frac{\beta^{t+1}}{\sum\limits_{i=0,N} \nabla_{\widehat{X}}\widehat{J}(l\tau)^t \nabla_{\widehat{X}}\widehat{J}(l\tau)}. \tag{3.8}$$

Internal diagnostics will be performed on the basis of the identified model.

Internal diagnostics is carried out after study and understanding, i.e. first of all, it is necessary to identify a prototype system for each of the states of failure / deterioration of performance E_C; let X_C – be the corresponding vector *P* of parameters characterizing the identified model $\widehat{H}_C$ of the state $E_C, c = 0,\ldots,(c-1)$.

Secondly, each of the specified vectors is associated with a probability region proportional to the variation of the vector X_C.

Finally, during the operational phase, *N* previous discrete values of the output signal are recorded and then a vector *X* of system parameters *S* is identified. The vector *X* is classified into one of the classes $c = 0,\ldots,(c-1)$ by comparing it with recognized vectors X_C.

In order to improve the quality of identification, it would be correct to use the same test signal $e(\cdot)$ with duration $N\tau = T$ for systems in the various states discussed above.

Let's now consider the communication channel diagnostics procedure implemented in the *CD* module of the M6 model [76].

Let's consider the basic provisions of diagnostics of continuous diagnostics systems.

Let's ssume that the considered system *S* can be represented by a transition operator *H* that will transform a multidimensional input signal *e(t)* into a multidimensional output signal *j(t)*. According to the temporary functions, denoted $e(\cdot)$ and $j(\cdot)$ will be stochastic processes in the dynamic case or probabilistic variables in the static case. The previous representation assumes no

feedback loop returning the signal *j(t)* to the system *S* input; and if not, then in the system from the very beginning of the looped feedback, let's assume the presence of a transformation that gives its description by *S* of the transformed open-loop system. The hypothesis is proposed that the input signal *e(t)* and the output signal *j(t)* available for observation and measurement at any moment of time and with the help of appropriate sensors. In the extreme case, it will be necessary to install a certain number of sensors specific to each component or subsystem.

Any pair $(e(\cdot), j(\cdot))$ is called a signature of a subsystem S, where $j(\cdot)$ is the output that the system gives to the input signal $e(\cdot)$. In this case, in the set of functions $e(\cdot)$ and $j(\cdot)$ accordingly, let's define two semimetrics d_e and d_j, that is, two metrics whose kernel is not zero:

$$\exists e_1(\cdot), e_2(\cdot)\ ,\ e_1(\cdot) \neq e_2(\cdot)\ ,\ d_e(e_1 e_2) = 0\ ,$$

$$\exists j_1(\cdot), j_2(\cdot)\ ,\ j_1(\cdot) \neq j_2(\cdot)\ ,\ d_j(j_1 j_2) = 0\ .$$

Let's also specify a finite collection *C* of different signatures $(e_c(\cdot), j_c(\cdot)), c = 0,\ldots,(c-1)$, characterizing *C* possible states E_C system *S* of the given design and established initial standard. Each of the pairs represents a type of impairment (decreased level of protection) E_C of the system *S.*

System *S* will be characterized by performance degradation E_C, $c = 0,\ldots,(c-1)$, if (and only if) its input $e(\cdot)$ and the corresponding output $j(\cdot)$ simultaneously satisfying:

$$d_e(e, e_c) = 0 \text{ and } d_j(j, j_c) = 0.$$

It is quite clear that among the *C* possible states of the system *(S)* there is a state E_0 of the system, called "the system in working order", which meets all established technical and operational standards. State E_0 is represented by a signature $(e_0(\cdot), j_0(\cdot)$.

It follows from the preliminary formal diagnosis that the specified problem is equivalent to the classification of internal states *x(t, e*) of the systems *S.* State *x(t, e*) is called observable if (and only if) it is possible to calculate it at a moment in time τ based on knowledge *(e(t),j(t))* at $t > \tau$. If one of the internal states $x_c(\cdot, e_c)$ is not observable, then the type of fault E_C, $c = 0,\ldots,(c-1)$ cannot be automatically diagnosed. Thus, let's believe that states E_C, which characterize the deterioration of working capacity, are associated with the observed internal states $x_c(\cdot, e_c)$.

In this case, it will be formally equivalent to characterize the deterioration of working capacity E_C, $c = 0,\ldots,(c-1)$ by signature $(e_c(\cdot), j(\cdot))$ or a corresponding internal state $x_c(\cdot, e_c)$, or the transition operator $H_c = f(x_c(\cdot, e_c))$, which depend on $x_c(\cdot, e_c)$. This value of semimetric d_e, d_j must correspond, thus, the semimetric d_x between the internal states of the system *S*, which is also called the structural distance. In particular, the system *S* will be considered a malfunction having (a decrease in the level of channel security) if:

$$d_x(x(\cdot, e), x_c(\cdot, e_c)) = 0. \tag{3.9}$$

Types of impairment of working capacity E_C, $c = 0,\ldots,(c-1)$ can be diagnosed with the help of the above hypotheses in two ways: external diagnostics and internal diagnostics.

External diagnostics use the signature $(e(\cdot), j(\cdot))$ without clear observation, measurement and direct assessment of the parameters characterizing the internal state of the system (S); requires external research data, namely: signatures $(e_c(\cdot), j_c(\cdot)), c = 0,\ldots,(c-1)$ and semi-distances d_e, d_j and often uses transient analysis for diagnostic purposes.

Internal diagnostics use a signature $(e(\cdot), j(\cdot))$ for observation, measurements or evaluation of parameters characterizing the internal state of the system: $S(\cdot) = H(e(\cdot)), H = f(x(\cdot, e))$. Thus, internal diagnosis is similar to the problem of identification and requires internal research data. Let's distinguish two cases:

1. Test diagnostics: in this case, a known input signal is applied to the S system $e(\cdot)$, and diagnosis is started only after the time has expired T; the data that must be registered for the classification of malfunctions is: $\{(e(t), j(t), t \in [0,T]\}$; the test requires the termination of normal operation.

2. Adaptive diagnostics: performed at any time $t \geq 0$ in the sense that the main problem is the detection of sudden changes in the internal state of the S system or deterioration of performance exceeding the established thresholds; data that must be registered for the classification of malfunctions at the moment of time $t \geq 0$, is $\{(e(\tau), j(\tau), \tau \in [0,T]\}$.

For systems that continuously transmit information, the most acceptable type is adaptive internal diagnostics. It is quite logical to include a self-diagnosis component in such a system, including monitoring the state of the data transmission channel environment. Based on the received diagnostic data, the transmission system adjusts the protection parameters (encryption, noise reduction, etc.). Thus, there is a continuous process of self-diagnosis with the subsequent change in the characteristics of the operating system without stopping it. Based on this, the information transmission system will turn into a feedback system, where the transmission channel affects the response of the system.

Thus, for a complete description of the composition of the M6 model, in the general case, it is necessary to accept the M5 model as the original, taking into account the changes made to it (for the reasons described above):

1. A new component (*ID*) is introduced into the model, which is responsible for the internal diagnostics of the IPS.

2. For the purpose of close interaction of the IPS and the information transmission system, the *CD* component – the communication channel diagnostic module – is included in the model.

3. As a result of the appearance of two new components of the model, new technological connections arise: C_{ik} – diagnostic relationship between *ID* and IPS; C_{ck} – communication that provides information about the state of the communication channel for the IPS using the *CD* module.

4 RESEARCH AND SIMULATION OF THE MECHANISM OF VULNERABILITIES VALIDATION IN ACTIVE ANALYSIS OF INFORMATION NETWORK SECURITY

ABSTRACT

A technique for analyzing the quality of the mechanism for validating the identified vulnerabilities of a corporate network has been developed, which is based on integral equations that take into account the quantitative characteristics of the mechanism for validating vulnerabilities under study at a certain point in time. This technique allows to build the laws of distribution of quality indicators of the vulnerability validation process and quantify the quality of the mechanism for validating detected vulnerabilities, which allows to monitor and control the validation of identified vulnerabilities in real time during active security analysis.

A method is proposed for constructing a fuzzy knowledge base for making decisions when validating vulnerabilities of software and hardware platforms with an active analysis of the security of a target corporate network based on the use of fuzzy logic, which makes it possible to provide reliable information about the quality of the mechanism for validating vulnerabilities indirectly.

The constructed knowledge base allows to form decisive decision-making rules for the implementation of a particular attacking action, which allows to develop expert systems to automate the decision-making process when validating the identified vulnerabilities of target information systems and networks.

The method of automatic active security analysis has been further developed, which, based on the synthesis of the proposed models, techniques and methods, allows, unlike the existing ones, to abstract from the conditions of dynamic changes in the environment, i.e. constant development of information technologies, and take into account only the quality parameters of the vulnerability validation process itself.

KEYWORDS

Cyber incident, vulnerability, warfare, risks, information security, electronic intelligence.

4.1 EXPERIMENTAL STUDY OF THE FUNCTIONING OF MODERN AUTOMATED VULNERABILITIES EXPLOITING MEANS

As mentioned earlier, modern systems of active security analysis (SASA) of information systems and networks, based on various methods of detecting and confirming vulnerabilities (in particular, methods of conducting penetration testing), allow simulating a potential cyber attack on the organization's information infrastructure and establishing its actual state security.

At the same time, the generalized algorithm of such systems consists of the following steps [77–95]:

– scanning of the target network, which allows to determine the list of available hosts, detect open ports on them and identify running services;

– making assumptions about the presence of vulnerabilities in the software, in particular in the detected services, based on the vulnerabilities knowledge base, errors in the configuration of equipment and other gaps in the protection perimeter of the information infrastructure;

– verification and confirmation of the possibility of implementing identified vulnerabilities by attempting to exploit them using specialized software, in particular, using so-called vulnerability exploits (malicious scripts, executable modules, etc.). This process is schematically shown in **Fig. 4.1**;

– generation of a report, which necessarily includes a list of validated vulnerabilities and their level of criticality (danger), as well as, optionally, recommendations for eliminating these vulnerabilities.

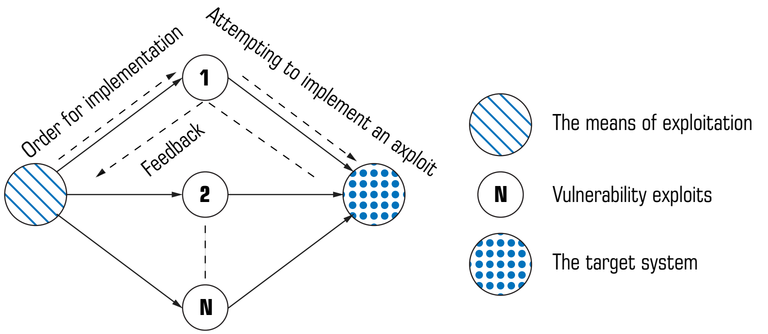


Fig. 4.1 The process of selecting and implementing the next exploit, with the subsequent recall of the target

In general, the need to check the possibility of implementing detected vulnerabilities, that is, their validation, arises because security scanners allow detecting only potential vulnerabilities of target systems, while allowing for the fallacy of such activations, which consists in the impossibility of actual implementation of the detected vulnerability on the part of the attacker. And since each SASA, having its own databases of ready-to-use exploits of known vulnerabilities and algorithms for their automatic implementation, the validation of detected vulnerabilities is carried out differently. So, for example, such algorithms can be based on the sequential implementation of all exploits available in the database, or taking into account simple criteria, such as the family of the operating system, the service and the rank of the exploit.

Thus, even in automated SASA, the number of vulnerability checks of only one target system, that is, attempts to exploit them, can reach several thousand (the main limitation is the number of

exploits in the database). This can be critical from the point of view of the time required to conduct an active security analysis in corporate networks, moreover, the sequential launch of all available exploits significantly increases the risk of a critical error in the functioning of the target system and its complete failure. Accordingly, in view of these limitations, it becomes expedient to solve the problem of determining the quality of the mechanism for validating vulnerabilities of software and hardware platforms.

For this, first of all, by processing the results of a large number of observations of the functioning of the means of exploiting the identified vulnerabilities, in particular, the feedback scheme shown in **Fig. 4.2**, the general characteristics of the vulnerability validation process, which take into account the above factors, were highlighted.

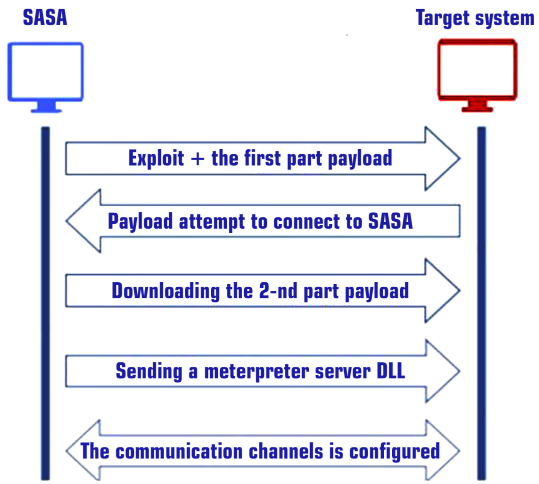


Fig. 4.2 Generalized scheme of step-by-step loading of the command interpreter – "meterpreter"

The feedback scheme (**Fig. 4.2**) is demonstrated on the example of the step-by-step implementation of the exploit with the previously mentioned payload, namely, the meterpreter command interpreter. At the same time, interaction (connection establishment) at the transport level takes place in accordance with the TCP protocol (**Fig. 4.3**) [84].

At the first stage, the exploit is transmitted to the target system along with the backed-up first part of the payload. After exploiting the vulnerability, the payload tries to connect back to the active security analysis system (i.e., the exploit tool, in this case metasploit) and establish a communication channel.

The next stage involves loading the second part of the payload DLL (Dynamic-link library – dynamically linked library) of the injection, after its successful execution, the exploit tool sends the DLL to the meterpreter server to establish the proper communication channel.

Thus, if the selected exploit, as well as the corresponding payload, worked and an active access session to the target system was obtained, it is possible to speak of a successful validation of the vulnerability.

Otherwise, when the selected exploit did not work, the vulnerability is not validated. If the exploit tool did not receive a response from the target system within the specified RTD delay interval and lost communication with it, the attempt to launch the selected exploit was unsuccessful and resulted in a critical error in the target system.

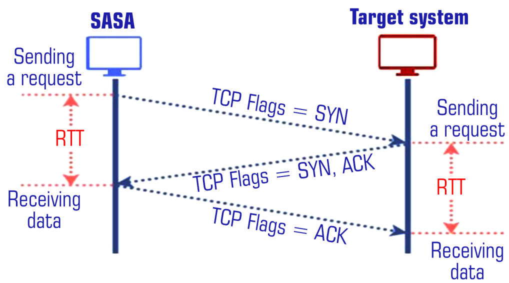


Fig. 4.3 The scheme of using the socket interface to establish a TSR connection

Thus, it was established that the quality of host vulnerability validation of the target corporate network is determined by the vector (q_s, q_f, q_c) of the three-dimensional vector space [96], where q_s – abscissa, which defines the number of successfully validated vulnerabilities, q_f – ordinate, which defines the number of unvalidated vulnerabilities and q_c – an application that determines the number of cases of vulnerability validation that resulted in critical errors on the target host and subsequent loss of communication with it.

However, given the risk of causing critical errors in the functioning of the target systems, their number in the corporate network, the dynamics of changes in the target systems themselves (changes in their configurations), as well as the constant increase in the number of new vulnerabilities and their exploits, it becomes difficult, experimentally, to ensure a full check of all objects (in this case, attempts to exploit vulnerabilities) related to this problem, in order to obtain a general population.

Therefore, it was decided to use a sample population in order to search for and study regularities in the process of active analysis of the security of corporate networks.

And since the sample is a subset of the general population, on the basis of its research using the tools and methods of mathematical statistics, it will be possible to draw a conclusion about the properties of the validation process that occurs in the general population.

At the same time, in order for the conclusions regarding the properties of the general population, which are made during the study of the sample, to be justified, it is necessary that this sample population is necessarily representative, that is, it accurately reflects the general population.

In general, the issue of forming a sample population is the first step on the way to obtaining results that objectively reflect the processes and phenomena that take place in the general population. And since the most common practice is to first select the required number of objects, and only then conduct their research, a list of 11 target systems was formed.

At the same time, the platforms were chosen on the basis of statistical data from Netmarketshare and Statcounter companies (**Fig. 4.4**) regarding the prevalence of the use of specific operating systems in the world [77–89] and in particular in Ukraine [89]. Also, two specially designed platforms for conducting penetration testing with known vulnerabilities already present were included in the list.

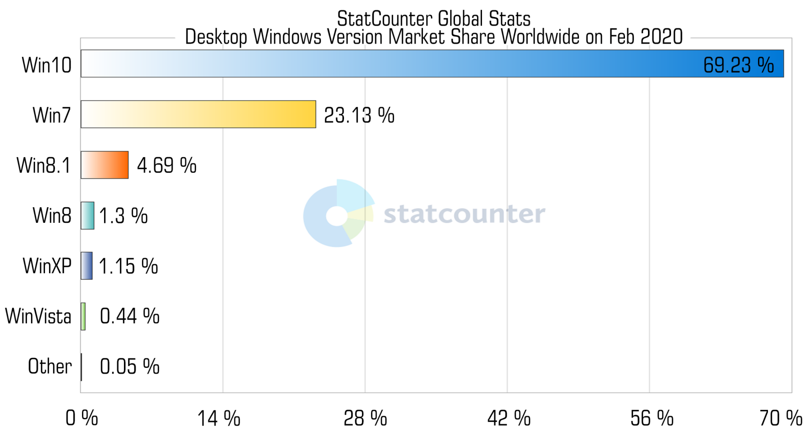


Fig. 4.4 Statistical data on the use of operating systems in the world

The experiment itself, in order to obtain the so-called functional dependencies, was carried out on a specially developed test bench, according to the proposed methodology of experimental research on the functioning of modern automated means of exploiting vulnerabilities, and the results were designed and presented in the form of a table (**Table 4.1**).

Table 4.1 Results of vulnerability validation using armitage and autopwn

Platform (OS)	Armitage					Metasploit-autopwn				
	£	q_s	q_f	q_c	t	£	q_s	q_f	q_c	t
Windows XP SP2	312	3	306	3	345	63	3	58	2	244
Windows XP SP3	98	3	93	2	86	58	3	53	2	286
Windows 7	85	2	80	3	65	63	3	60	2	369
Windows 8.1	83	1	81	1	58	65	0	64	1	281
Windows 10	84	0	83	1	154	1255	0	1255	0	1523
Windows Server 2008 R2	96	2	92	2	82	84	1	82	1	363
Windows Server 2016	39	0	39	0	71	32	0	32	0	43
Mac OS X 10.13	63	1	61	1	115	59	1	58	0	249
Mac OS X 10.14	46	1	45	0	83	41	1	40	0	58
Metasploitable 2	765	3	762	0	293	1445	3	1442	0	1462
Metasploitable 3	780	3	777	0	330	1911	3	1908	0	1933

where £ – the total number of attempts to exploit detected vulnerabilities of a separate host of the target corporate network; t – total validation time of detected vulnerabilities of a separate host of the target corporate network, expressed in seconds

4.1.1 TEST STAND DESCRIPTION

All experiments were performed on a machine running Windows 10 Pro x64 v1803 operating system with an Intel Core i5-3210M CPU 2.50 GHz and 12 GB RAM using the VMware Workstation 12 Pro v12.5.9 build-7535481 virtualization platform, on which a special test stand was deployed. The schematic representation of this stand is presented in **Fig. 4.5**.

Virtual machines running the Kali GNU / Linux Rolling 2019.3 OS with the following tools for automating the work of the Metasploit framework vulnerability exploitation tool is installed and configured as a system of automatic active security analysis (SAASA) in various experiments:

– metasploit-autopwn;

– armitage.

As a target host, a number of virtual machines with different installed platforms and the corresponding standard set of software act:

– MS Windows 10;

– MS Windows Server 2008 R2;

– MS Windows Server 2016;

– Mac OS X 10.13 and 10.14;

– Metasploitable 2 and 3.

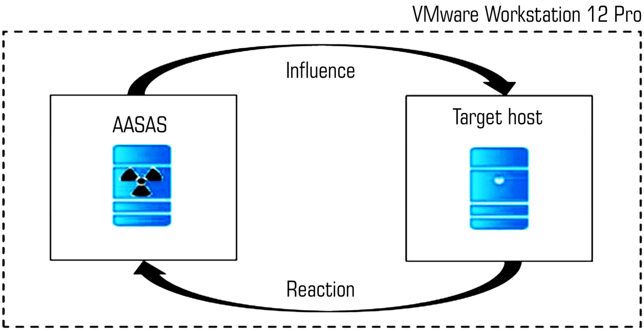


Fig. 4.5 Generalized scheme of the test stand

4.1.2 METHODOLOGY OF EXPERIMENTAL STUDY OF THE FUNCTIONING OF MODERN AUTOMATED MEANS OF EXPLOITING VULNERABILITIES

First of all, it should be noted that all experiments consist in conducting automatic active security analysis of a number of the same target hosts using various automated software tools of active security analysis defined in the previous subsection and further analysis of their work results.

The purpose of this experimental study is to determine the general characteristics of the vulnerability validation process. For this purpose, the following method of experimental research is proposed, where the following system of actions is provided:

1. After deployment of the test bench and configuration of all target hosts, create snapshots (VMware snapshot) [97–112] of data of virtual machines to save their original (initial) state. A virtual machine snapshot is a point-in-time copy of the virtual machine disk file (VMDK) that allows to restore the saved state of the virtual machine.

2. Conduct an analysis of the security of the next host using the Armitage graphical cyber attack management tool using the Hail Mary vulnerability exploitation mode, save the results and restore the initial state of the target host under investigation with VMware tools.

3. Carry out an analysis of the security of the next host using the db_autopwn automatic exploit and cyber attack plugin, save the results and restore the initial state of the target host under investigation with VMware tools.

4. If a critical error occurs in steps 2 and 3 during active security analysis, restore the initial state of the target host under investigation, and re-analyze it with exclusion from the list of exploits that led to this error.

5. Submit the results of the conducted experiments in the form of a table.

4.2 MATHEMATICAL MODELING OF INFORMATION SYSTEMS AND NETWORKS IDENTIFIED VULNERABILITIES VALIDATION MECHANISM

4.2.1 REGRESSION ANALYSIS OF EXPERIMENTAL RESEARCH RESULTS

On the basis of the conducted experimental studies, it was established that each of the coordinates of the vector on the one hand, it changes continuously over time, during which an active analysis of the security of an individual target host and the corporate network as a whole is carried out, and on the other hand, all three coordinates are connected to each other by some functional dependence.

However, unlike deterministic dynamic systems, which can be described by systems of differential equations built on the basis of the system`s nature, the task of detecting vulnerability validation is not unambiguous. Therefore, it was decided to solve the task of building a mathematical model of information systems and networks vulnerabilities validation mechanism by means of regression analysis [78, 87, 91], creating analytical dependencies, which in turn are solutions of some differential equations system.

In general, regression analysis refers to the study of the regularity of the relationship between two variables, when one x value corresponds to a set of y values, i.e. the relationship between them is not fully defined [88].

Thus, in regression analysis, statistical dependencies are described by a mathematical model, that is, a regression equation that reproduces the relationship between factor values and variable characteristics of the investigated process establishing the corresponding analytical dependence, and has the form $y = f(x)$.

At the same time, the regression equation, if possible, should be quite simple and adequate.

The analysis itself is carried out directly in several steps:

– checking for the presence of a correlation relationship;

– approximation of experimental data;

– statistical analysis of regression equations.

First of all, the statistical relationship between two variables is evaluated based on the results of experimental observations using the correlation coefficient.

Provided that N observations are made, resulting in two samples:

$$x_1, x_2, \dots, x_n, \quad y_1, y_2, \dots, y_n,$$

the correlation coefficient is determined by the following formula:

$$R = \frac{\sum_{i=1}^{n}\left(x_i - \tilde{x}\right)\left(y_i - \tilde{y}\right)}{\left(n-1\right)\sigma_x\sigma_y}, \tag{4.1}$$

where $\tilde{x}, \tilde{y}$ – the mean value of the sample *X* and *Y*, respectively, which establishes the center of the sample population and is determined by formulas:

$$\tilde{x} = \frac{1}{n} \cdot \sum_{i=1}^{k} x_i, \quad \tilde{y} = \frac{1}{n} \cdot \sum_{i=1}^{k} y_i, \tag{4.2}$$

σ_x, σ_y – the mean squared deviation for *X* and *Y*, respectively, is defined as the square root of the sample variance:

$$\sigma_x = \sqrt{D_x}, \quad \sigma_y = \sqrt{D_y}, \tag{4.3}$$

D_x, D_y – sample variance, which characterizes the variability of the values in the sample of *X* and *Y*, respectively, that is, the variation of observations, and is determined by the formulas:

$$D_x = \frac{1}{n-1} \cdot \sum_{i=1}^{k} \left(\tilde{x} - x_i\right)^2, \quad D_y = \frac{1}{n-1} \cdot \sum_{i=1}^{k} \left(\tilde{y} - y_i\right)^2. \tag{4.4}$$

The expression $(n-1)$ from formulas (4.4) is called the number of degrees of freedom. This number is equal to the number of independent values involved in determining any parameter of a statistical population. When determining the variance, one degree of freedom is spent on determining the average value [90].

It should be noted that the value of the correlation coefficient is always within the limits $-1 \le R \le 1$. At the same time, it characterizes only a linear relationship between random variables. That is, with a positive value of the coefficient, it can be assumed that when one value increases, the other also increases on average, and with a negative value, on the contrary, the growth of one value leads to a decrease in the other value on average. The closer the value *R* to $+1$ or -1, the closer the linear relationship between the *x* and *y* values, however, if the value $R = 0$, this indicates its absence. In general, a satisfactory value of relationship density is considered to be $R \ge 0{,}5$, good at $R = 0{,}8 \ldots 0{,}85$.

Verification of the correspondence of the sample value of the correlation coefficient *R* to the correlation value (ρ) between general populations *x* and *y*, occurs with the use of *t* – distribution of Student [112]. For this, the calculated value t_{est} is first found according to formula (4.5) and compared with the table (**Table 4.2**).

$$t_{est} = |R| \sqrt{\frac{n-2}{1-R^2}}. \tag{4.5}$$

If $t_{est} > t_{tab}$ with the number of degrees of freedom $f = n - 2$ and significance level $\alpha = 5~\%$, then the correlation relationship exists and is confirmed for general populations.

Table 4.2 Student's test value for significance $\alpha=0.05$

f	1	2	3	4	5	6	7	8	9	10	60
t_{tab}	12,71	4,303	3,182	2,775	4,571	2,447	2,305	2,228	2,086	2,042	2,00

The next step, in order to present in an understandable and concise form the empirical dependencies between the parameters describing the behavior of the system, is the approximation of the experimental data.

In general, approximation is the process of constructing an approximate (approximating) function based on the results of experimental studies that passes through all points of the initial data and is closest to a given continuous function. The process itself consists of two main stages [87]:

1) the selection of the general form of a typical functional dependence (approximants), which is carried out either for theoretical reasons or with the help of a graphical representation of the results of the experiment, analyzing the location of the points (x_n, y_n) on the Cartesian coordinate plane. At the same time, the values of the factor are placed on the abscissa axis, respectively, the values of the evaluation parameter are placed on the ordinate axis, and the actual results are indicated by dots. By directly connecting these points with a straight line, let's obtain a graph of the results of the conducted experiment, and by drawing another straight line (curve) through the middle points of each of the obtained segments, there is an approximate representation of the graph of the desired approximant. After that, the obtained graph is compared with the graphs of typical functions and the general appearance of the approximating function is selected, which will most similarly describe the investigated dependence;

2) determination of the best numerical values of the parameters (coefficients) of the approximant.

4.2.2 MATHEMATICAL METHODS OF FUNCTION APPROXIMATION

In general, the need to approximate a function by some analytical model arises when solving a fairly wide range of tasks, in particular tasks of statistical radio engineering and radio physics, control theory and data transmission systems [105, 112–114].

For example, approximation tasks are solved when modeling acoustic signals and designing telecommunication systems [103, 104], mobile communication systems [102], when optimizing the parameters of the reversible (lossless) digital data compression procedure [113], as well as when creating mathematical models of control systems [31] and simulation of processes under the influence of random factors [80].

All methods of function approximation can be divided into two groups: parametric and non-parametric. The first use a priori information about the type and / or parameters of the general distribution. Non-parametric ones, in turn, work with greater uncertainty regarding a priori information, including even its complete absence, and therefore have a much wider scope of application.

However, non-parametric methods, in comparison with parametric ones, are more time-consuming from the point of view of performing mathematical calculations.

It should be noted that currently, one of the main approaches to solving nonparametric approximation problems is polynomial estimation based on the first theorem of *K*. Weierstrass, which is as follows: if the function $f(x)$ is continuous on a segment $[a,b]$, then there exists a sequence of polynomials $\{Pn(x)\}$, which uniformly on the segment $[a,b]$ converges to $f(x)$, that is, for any $\varepsilon > 0$ polynomial will be found $P_n(x)$ with the number *n* depending on ε, such that $|P_n(x) - f(x)| < \varepsilon$, at once for all *x* from the interval $[a,b]$.

This theorem was proved in 1912 by the famous Soviet scientist S. N. Bernshtein [4, 5]. As approximating polynomials $P_n(x)$, polynomials of the following form were used:

$$B_n(f;x) = B_n(x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) b_{k,n}(x), \qquad (4.6)$$

where $b_{k,n}(x) = C_n^k x^k (1-x)^{n-k}$, $C_n^k = n!/\,k!(n-k)!$.

Function $b_{k,n}(x)$ are called the basis Bernstein polynomial of degree *n*, operators $B_n(f;x)$ respectively, Bernstein polynomials of order *n* functions $f(x)$, and the coefficients $f(k/n)$ – Bernstein coefficients.

S.N. Bernstein, relying on elementary results from the theory of probabilities, proved that the sequence of polynomials $\{B_n(f;x)\}$ at $n \to \infty$ converges to $f(x)$ uniformly on $[0,1]$, that is

$$\lim_{n\to\infty} \|f - B_n(f)\| = 0.$$

Thus, the following theorem holds.

Theorem 2.1. If the function $f(x)$ on a segment $[0,1]$ satisfies the Lipshitz condition [31] with a constant M, then with every $n \geq 2$ and every $x \in [0,1]$, a fair estimate

$$|B_n(f;x) - f(x)| \leq M\sqrt{\frac{x(1-x)}{n}}. \qquad (4.7)$$

4.2.3 MATHEMATICAL MODEL OF ANALYSIS OF VULNERABILITY VALIDATION PROCESS QUANTITATIVE CHARACTERISTICS

Based on the results of an experimental study of the functioning of modern automated means of exploiting vulnerabilities obtained in 4.1 (**Table 4.1**), let's build a mathematical model for the analysis of quantitative characteristics of the process of validating information system vulnerabilities by means of regression analysis. To do this, let's first estimate the statistical relationship

between variables t and q_s, q_f, q_c, obtained during the study of the validation mechanism of the Armitage cyber-attack management graphic tool, using the correlation coefficient (4.1).

According to the data in **Table 4.1**, let's calculate the auxiliary values: the average value of the sample (4.2), the sample variance (4.4) and the mean squared deviation (1.3). Let's summarize the results in **Table 4.3**.

Table 4.3 Estimated values of the correlation coefficient

Searched values	**Armitage**		
	t, q_s	t, q_f	t, q_c
$\tilde{t}$	152,91	152,91	152,91
$\tilde{q}$	1,73	219,91	1,18
D_t	12716,09	12716,09	12716,09
D_q	1,42	79027,89	1,36
σ_t	112,77	112,77	112,77
σ_q	1,19	281,12	1,17
R	0,6	0,84	-0,07

In addition, let's also check the correspondence of the sample value of the correlation coefficient R to the correlation value (ρ) between general aggregates of quantities t and q_s, q_f, q_c using Student's distribution. Let's find the value t_{est} by formula (4.5). The results are presented in **Table 4.4**.

Table 4.4 Criterion of the correlation coefficient significance

Calculated value	**Armitage**		
	t, q_s	t, q_f	t, q_c
t_{est}	2,254	4,693	0,216

Comparing the obtained values t_{est} with theoretical ones (**Table 4.1**) with the number of degrees of freedom $f = n - 2 = 9$ and significance level $\alpha = 5\%$ there are the following results:

– for a pair t, q_s – $t_{est} > t_{tab}$ since $2{,}254 > 2{,}096$, this indicates that there is a direct relationship between the time of validation of detected vulnerabilities and the number of successfully validated vulnerabilities;

– for a pair t, q_f – $t_{est} > t_{tab}$ since $4{,}693 > 2{,}096$, this indicates that there is a significant direct relationship between the time of validation of detected vulnerabilities and the number of unvalidated vulnerabilities;

– for a pair t, q_c – $t_{est} < t_{tab}$ since $0{,}216 < 2{,}096$, this indicates that there is a very weak relationship between the time of validation of detected vulnerabilities and the number of cases of validation of vulnerabilities that lead to critical errors on the target host, however, based on previous results, it can be argued that this situation is solved by increasing the number of observations.

Next, in order to present the empirical dependences between the parameters in a clear and concise form, let's approximate the experimental data. To do this, first, let's find out the class of functions to which the desired approximant belongs by constructing a graph of the experiment results and an approximate graph of the desired approximant for each of the pairs of variables (**Fig. 4.6**).

Fig. 4.6 shows that the functions have a polynomial representation and, at the same time, a value R^2 (reliability of the approximation) testify to the accuracy of the description of the initial dependence of the experimental data by the approximating function.

That is why, in order to obtain the most reliable coefficients of the approximant, let's use Bernstein's theorem.

From the data in **Table 4.1**, it can be seen that the time of the rational cycle of vulnerability validation, in the case of using the Armitage tool, is 345 seconds. Therefore, first it is possible to carry out normalization of the time segment $[0;345]$, as follows:

$$t_n = \frac{t_i}{T}, \tag{4.8}$$

where t_n – is the normalized time; T – target host vulnerability validation time in seconds (rational cycle time); t_i – the time for which the relevant characteristics (q_s, q_f, q_c) assumed their values within the rational cycle.

The results of normalization of the time segment are presented in **Table 4.5**.

Table 4.5 Normalization of the rational cycle time

real time – t	0	58	65	71	82	83	86	115	154	293	330	345	0
normalized time – t_n	0	0,168	0,188	0,206	0,238	0,241	0,249	0,333	0,446	0,849	0,957	1	0

Then the values of the variables $q_s(t_n)$, $q_f(t_n)$, $q_c(t_n)$, as functions of normalization time, presented in **Table 4.6**.

After that, using data from **Table 4.6** and representation (4.6), initial analytical dependencies for the number of successfully validated vulnerabilities were obtained $q_s = q_s(t_n)$.

$$q_s(t_n) = q_s(0)b_{0.11}(t_n) + q_s(0{,}168)b_{1.11}(t_n) + q_s(0{,}188)b_{2.11}(t_n) + q_s(0{,}206)b_{3.11}(t_n) +$$
$$+ q_s(0{,}238)b_{4.11}(t_n) + q_s(0{,}241)b_{5.11}(t_n) + q_s(0{,}249)b_{6.11}(t_n) + q_s(0{,}333)b_{7.11}(t_n) +$$
$$+ q_s(0{,}446)b_{8.11}(t_n) + q_s(0{,}849)b_{9.11}(t_n) + q_s(0{,}957)b_{10.11}(t_n) + q_s(1)b_{11.11}(t_n).$$

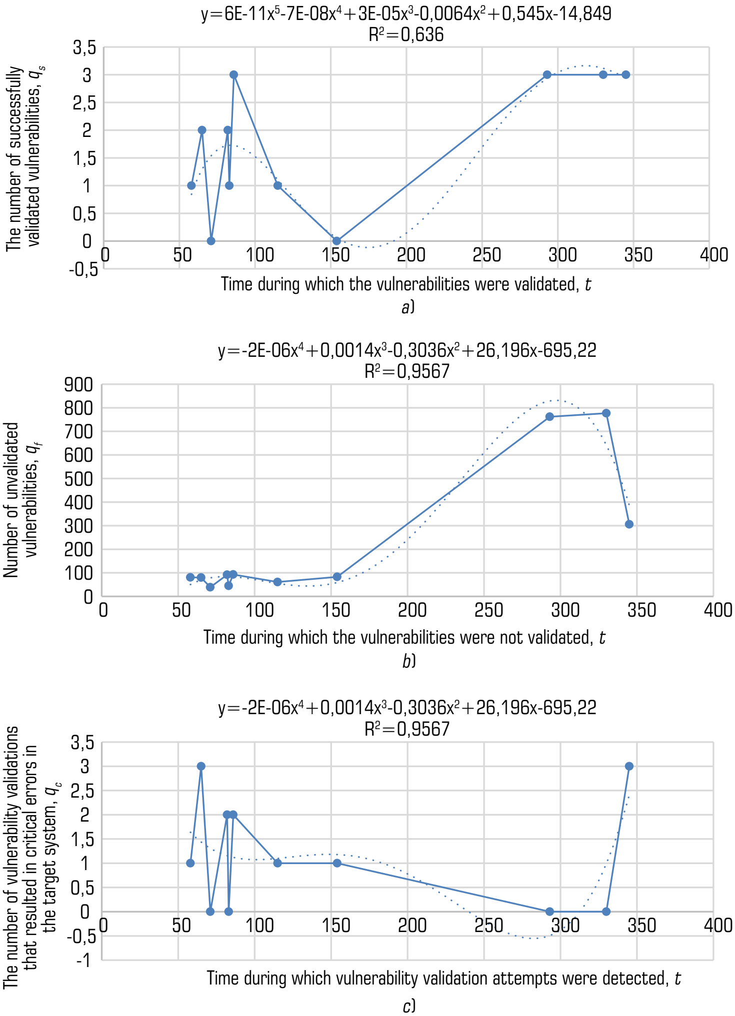


Fig. 4.6 Approximate graphs of the sought approximants for each of the pairs of variables: a – t, q_s; b – t, q_f; c – t, q_c

Table 4.6 Value of number of successfully validated $q_s(t_n)$, unvalidated vulnerabilities $q_f(t_n)$ and cases of validations that led to critical errors $q_c(t_n)$

t_n – normalized time	0	0,168	0,188	0,206	0,238	0,241	0,249	0,333	0,446	0,849	0,957	1	0
$q_s(t_n)$	0	1	2	0	2	1	3	1	0	3	3	3	0
$q_f(t_n)$	0	81	80	39	92	45	93	61	83	762	777	306	0
$q_c(t_n)$	0	1	3	0	2	0	2	1	1	0	0	3	0

Table 4.7 The value of polynomials $b_{k,11}(t_n)$

k	$b_{k,11}(t_n)$
0	$(1-t)^{11}$
1	$11t(1-t)^{10}$
2	$55t^2(1-t)^9$
3	$165t^3(1-t)^8$
4	$330t^4(1-t)^7$
5	$462t^5(1-t)^6$
6	$462t^6(1-t)^5$
7	$330t^7(1-t)^4$
8	$165t^8(1-t)^3$
9	$55t^9(1-t)^2$
10	$11t^{10}(1-t)$
11	t^{11}

After substituting the corresponding values from **Tables 4.6** and **4.7**, simplifying the expression, there is

$$q_s(t_n) = b_{1.11}(t_n) + 2b_{2.11}(t_n) + 2b_{4.11}(t_n) + b_{5.11}(t_n) + 3b_{6.11}(t_n) + b_{7.11}(t_n) + \\ +3b_{9.11}(t_n) + 3b_{10.11}(t_n) + 3b_{11.11}(t_n). \quad (4.9)$$

After comparing the values of the calculation results and the data from **Table 4.6**, it follows (**Table 4.8**) that the deviations between the empirical and calculated data are permissible, and

when the number of values increases, these deviations become smaller and smaller. At the same time, it should be noted that for further research related to false vulnerability validation attempts and validation cases that led to critical errors, this difference is not significant.

Table 4.8 Comparative values for $q_s(t_n)$

t_n – normalized time	Empirical values $q^e_s(t_n)$	Calculated values $q^p_s(t_n)$	Deviation $\theta=\lvert q^e_s(t_n)-q^p_s(t_n)\rvert$
0	0	0	0
0,168	1	1,065446	0,065446
0,188	2	1,100162	0,899838
0,206	0	1,126111	1,126111
0,238	2	1,167208	0,832792
0,241	1	1,171013	0,171013
0,249	3	1,181262	1,818738
0,333	1	1,309026	0,309026
0,446	0	1,494756	1,494756
0,849	3	2,425641	0,574359
0,957	3	2,970647	0,029353
1	3	3	0

The dependence graph (4.9) is presented in **Fig. 4.7**, which shows that the function $q_s=q_s(t_n)$ of successful validation of vulnerabilities satisfies the Lipshitz condition [31], i.e., for arbitrary $t_n^{(1)},t_n^{(2)}\in[0;1]$ exist $K>0$, that the inequality is fulfilled

$$\left|q_s(t_n^{(1)})-q_s(t_n^{(2)})\right|\le K\left|t_n^{(1)}-t_n^{(2)}\right|. \tag{4.10}$$

It follows from the condition (1.10) that there is a rectangular region beyond which the graph of the function $q_s=q_s(t_n)$ does not step out.

This makes it possible to further build the laws of probability distribution of the number of successfully validated vulnerabilities. In addition, when condition (4.10) is fulfilled, estimate (4.7) is valid, i.e.

$$\left|B_n(q_s,t_n)-q_s(t_n)\right|\le K\sqrt{\frac{t_n(1-t_n)}{n}}. \tag{4.11}$$

It follows from the inequality (4.7) that there exists such a positive number K, at which

$$\theta = \left| q_s^e(t_n) - q_s^p(t_n) \right| = K\sqrt{\frac{t_n(1-t_n)}{n}}. \quad (4.12)$$

Dependence (4.12) makes it possible to set the appropriate precision for determining the power n of the Bernstein polynomial.

Thus, using data from **Table 4.8** and dependence (4.12), the maximum value was obtained K for $q_s = q_s(t_n)$:

$\max(k_i) = 13{,}949121$, where $i \in [1;11)$.

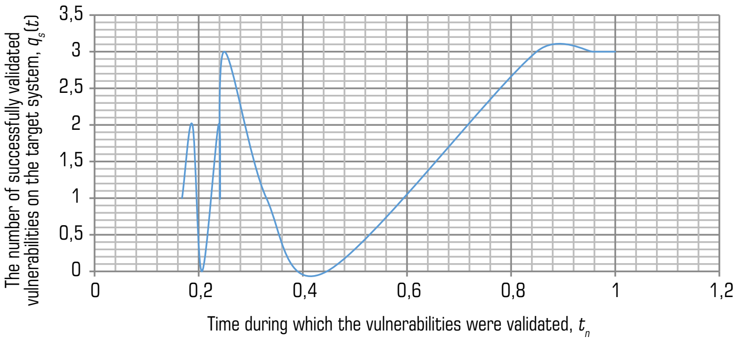


Fig. 4.7 The target system on the time of the rational cycle

Similarly, using representation (4.6), data from **Tables 4.6** and **4.7**, let's obtain initial analytical dependencies for the number of unvalidated vulnerabilities $q_f = q_f(t_n)$ (dependency (4.13), **Fig. 4.8**) and the number of cases of vulnerability validation that led to critical errors $q_c = q_c(t_n)$ (dependency (4.14), **Fig. 4.9**).

Tables 4.9 and **4.10** present the corresponding comparative values of the calculation results and data from **Table 4.6**.

$$q_f(t_n) = 81b_{1.11}(t_n) + 80b_{2.11}(t_n) + 39b_{3.11}(t_n) + 92b_{4.11}(t_n) + 45b_{5.11}(t_n) + 93b_{6.11}(t_n) + \\ +61b_{7.11}(t_n) + 83b_{8.11}(t_n) + 762b_{9.11}(t_n) + 777b_{10.11}(t_n) + 306b_{11.11}(t_n). \quad (4.13)$$

$$q_c(t_n) = b_{1.11}(t_n) + 3b_{2.11}(t_n) + 2b_{4.11}(t_n) + 2b_{6.11}(t_n) + b_{7.11}(t_n) + b_{8.11}(t_n) + 3b_{11.11}(t_n). \quad (4.14)$$

Also, it should be noted that **Fig. 4.8** and **4.9** shows that the functions $q_f = q_f(t_n)$ and $q_c = q_c(t_n)$ also satisfy the Lipshitz condition.

Table 4.9 Comparative values for $q_f(t_n)$

t_n – normalized time	Empirical values $q^e_f(t_n)$	Calculated values $q^p_f(t_n)$	Deviation $\theta = \lvert q^e_f(t_n) - q^p_f(t_n) \rvert$
0	0	0	0
0,168	81	62,547827	18,45217
0,188	80	63,809242	16,19076
0,206	39	64,596778	25,596778
0,238	92	65,508850	26,49115
0,241	45	65,575300	20,5753
0,249	93	65,743882	27,256118
0,333	61	67,844585	6,844585
0,446	83	78,745219	4,254781
0,849	762	538,115125	223,884875
0,957	777	478,499059	298,500941
1	306	306	0

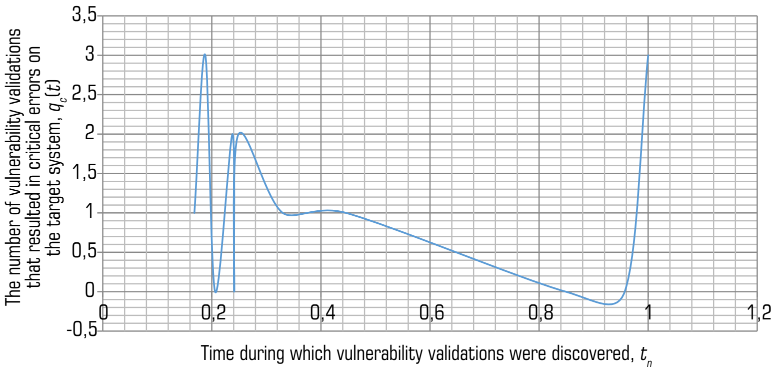


Fig. 4.8 Dependence of the number of unvalidated vulnerabilities of the target system on the time of the rational cycle

Table 4.10 Comparative values for $q_c(t_n)$

t_n – normalized time	Empirical values $q^e_f(t_n)$	Calculated values $q^p_f(t_n)$	Deviation $\theta=\lvert q^e_f(t_n)-q^p_f(t_n)\rvert$
0	0	0	0
0,168	1	1,337389	0,337389
0,188	3	1,360285	1,639715
0,206	0	1,364959	1,364959
0,238	2	1,346917	0,653083
0,241	0	1,343984	1,343984
0,249	2	1,335418	0,664582
0,333	1	1,221982	0,221982
0,446	1	1,125939	0,125939
0,849	0	0,731249	0,731249
0,957	0	1,860081	1,860081
1	3	3	0

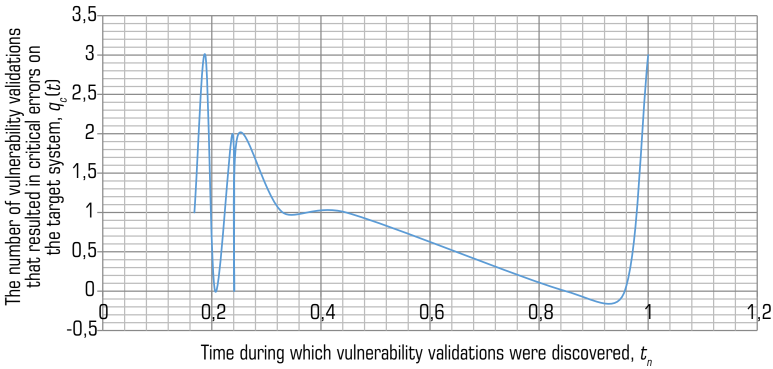


Fig. 4.9 Dependence of the number of validations of vulnerabilities that led to critical errors in the target system on the time of the rational cycle

In addition, using data from **Tables 4.9** and **4.10** and dependence (4.12), let's obtain the maximum K values for $q_f=q_f(t_n)$:

$\max(K_i)=4880{,}359905$, where $i \in [1;11)$,

and $q_c=q_c(t_n)$

$\max(K_i)=30{,}411511$, where $i \in [1;11)$.

Thus, as a result, analytical dependencies were obtained for the studied characteristics of the process of validation of vulnerabilities of information systems [19]:

$$q_s(t_n)=\sum_{i=0}^{n} q_s(t_n^{(i)})b_{k,n}(t_n),\ q_f(t_n)=\sum_{i=0}^{n} q_f(t_n^{(i)})b_{k,n}(t_n),\ q_c(t_n)=\sum_{i=0}^{n} q_c(t_n^{(i)})b_{k,n}(t_n). \quad (4.15)$$

4.3 METHODOLOGY FOR ANALYZING THE QUALITY OF WORK OF THE MECHANISM FOR CORPORATE NETWORK DETECTED VULNERABILITIES VALIDATING

Based on the practical analysis of the vulnerability validation process carried out in the previous section and the analytical dependencies of the basic characteristics of the vulnerability validation process (4.15) obtained with the help of Bernstein polynomials, it became possible to highlight and characterize additional key indicators that will allow more precisely determining the quality of the vulnerability validation mechanism, and also assert with high credibility about the positive progress or consequences of validating the vulnerabilities of the target corporate network.

As a result, the following quality indicators of the corporate network vulnerability validation mechanism were selected [95, 112–116]:

1) *A – accuracy* – the share of correctly made decisions regarding the implementation of specific exploits relative to all made decisions. This parameter characterizes the ability of the validation mechanism of detected vulnerabilities to successfully check and confirm the possibility of their implementation due to correctly made decisions regarding the use of selected exploits with the appropriate payload for these vulnerabilities;

2) *E – error* – the share of decisions made regarding the implementation of specific exploits that did not confirm the possibility of implementing the corresponding vulnerabilities in relation to all decisions made. The error parameter characterizes the ability of the mechanism of validation of detected vulnerabilities to make decisions regarding the use of selected exploits that do not work for certain reasons. Such reasons include, for example, the non-compliance of the target system with the conditions for implementing the selected exploit, changing the ports on which vulnerable services work by default, the reaction of the protection system – blocking the possibility of implementing the exploit;

3) *Ce – critical error* – the share of decision-making cases regarding the implementation of specific exploits, which led to critical errors in the target system and subsequent loss of

communication with it in relation to all decisions made. A critical error characterizes the ability of the mechanism of validation of detected vulnerabilities to make decisions regarding the use of selected exploits, which in the process of their implementation lead to a critical error in the functioning of the target system and its subsequent failure.

According to (4.15), these indicators are determined as follows:

$$A=\frac{\int_0^1 q_s(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.16}$$

$$E=\frac{\int_0^1 q_f(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.17}$$

$$Ce=\frac{\int_0^1 q_c(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}. \tag{4.18}$$

In addition, in order to evaluate the quality of the mechanism for validating detected vulnerabilities, taking into account all the above quality indicators, let's reduce expressions (4.16)–(4.18) into a single integral indicator:

$$J_{qv}=\frac{A}{E}-Ce, \tag{4.19}$$

where J_{qv} – integral index of the vulnerability validation mechanism quality.

At the same time, if $J_{qv}>1$, then the vulnerability validation mechanism has high quality.

Thus, it is possible to highlight the following steps of the methodology of quality analysis of the mechanism of validation of corporate network vulnerabilities [98]:

Step 1. Collection of statistical data regarding the process of validation of detected vulnerabilities of the corporate network of the evaluated validation mechanism.

Step 2. Normalization of the time segment of the vulnerability validation of the hosts of the target corporate network according to the expression (4.8).

Step 3. Construction of Bernstein polynomials to obtain initial analytical dependencies for basic characteristics (q_s, q_f, q_c) of the vulnerability validation quality.

Step 4. Calculation of more accurate performance indicators of the vulnerability validation mechanism: A – accuracy (4.16), E – error (4.17) and Ce – critical error (4.18).

Step 5. Evaluation of the performance of the mechanism for validating vulnerabilities of corporate networks based on the calculation of a single integral indicator (4.19).

Also, it should be noted that the dependencies (4.16)–(4.18) are generally functions of time

$$A(t_n)=\frac{\int_0^{t_n} q_s(\theta)d\theta}{\int_0^{t_n}(q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.20}$$

$$E(t_n)=\frac{\int_0^{t_n} q_f(\theta)d\theta}{\int_0^{t_n}(q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.21}$$

$$Ce(t_n)=\frac{\int_0^{t_n} q_c(\theta)d\theta}{\int_0^{t_n}(q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.22}$$

the graphs of which were constructed and displayed in **Fig. 4.10** on the basis of an experimental study of the mechanism of validation of information system vulnerabilities of the db_autopwn plugin (**Table 4.1**).

In addition, taking the derivatives of the distribution functions of the quantitative indicators of the quality of the mechanism of validation of detected vulnerabilities (4.20), (4.21) and (4.22), let's obtain the distribution densities, which are determined as follows:

$$\alpha(t_n)=\frac{dA(t_n)}{dt_n}, \tag{4.23}$$

$$\xi(t_n)=\frac{dE(t_n)}{dt_n}, \tag{4.24}$$

$$\varsigma(t_n)=\frac{dCe(t_n)}{dt_n}. \tag{4.25}$$

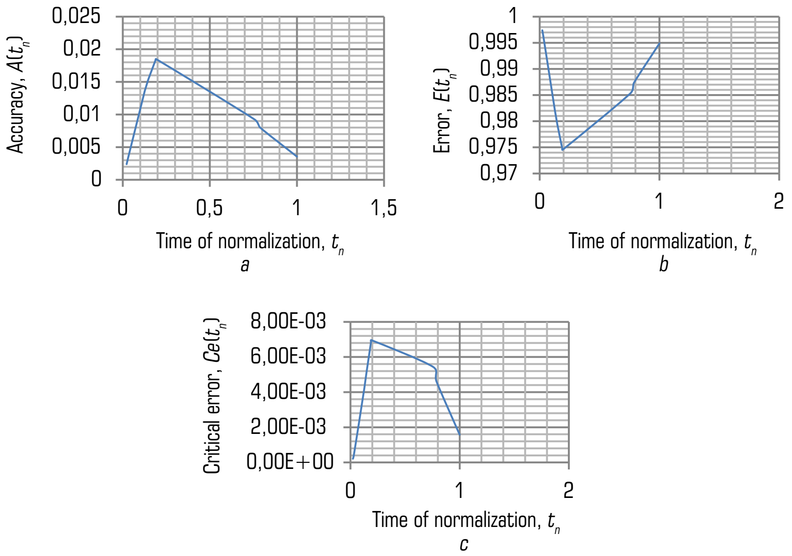


Fig. 4.10 Dependence of quantitative performance indicators of the vulnerability validation mechanism on the time of the rational cycle: *a* – accuracy $A(t_n)$; *b* – error $E(t_n)$; *c* – critical error $Ce(t_n)$

4.4 A METHOD OF CONSTRUCTING A FUZZY KNOWLEDGE BASE FOR DECISION-MAKING WHEN VALIDATING SOFTWARE AND HARDWARE PLATFORM VULNERABILITIES

Having analyzed the dependencies of the quality indicators of the corporate network vulnerability validation mechanism obtained in the previous subsection (**Fig. 4.10**), it can be seen that the maximum value of accuracy *A* takes the value 0,02 (a). At the same time, a minimal error *E* is within the limits from 0.97 to 0.98 (b), and the maximum critical error *Ce* is within the limits from $6 \cdot 10^{-3}$ to $8 \cdot 10^{-3}$ (c). This makes it possible to construct property functions for fuzzy sets, the elements of which are accuracy, error, and critical error.

Therefore, a decision was made to intellectualize the process of validating vulnerabilities of software and hardware platforms based on fuzzy technology [112], by creating a knowledge base for automatic decision-making when validating vulnerabilities during an active analysis of the security of corporate networks. This will make it possible to quickly, in real time, and with minimal risk, make appropriate decisions regarding attempts to implement specific exploits of vulnerabilities for their validation.

It should be noted that in order to build a knowledge base and further form decisive decision-making rules, first of all, it is necessary to select input and output parameters [80, 111].

Quantitative characteristics of the vulnerability validation process are used as input parameters, which include the number of successfully validated vulnerabilities $q_s(t)$, number of unvalidated vulnerabilities $q_f(t)$ and the number of instances of vulnerability validation that resulted in a critical error $q_c(t)$.

The output parameters are the distribution functions of quantitative indicators of the quality of the mechanism of validation of detected vulnerabilities (4.20)–(4.22).

As it was already established, from the conducted study of the vulnerability validation process, the initial parameters for building the knowledge base depend on the normalization time, which is a random variable that can take ambiguous values [112]. Based on this, with the use of statistical values obtained during the study of the mechanism of validation of information system vulnerabilities of the db_autopwn plugin (**Table 4.11**), the normalization time membership function was obtained (**Fig. 4.11**).

Table 4.11 The value of the number of successfully validated $q_s(t_n)$, unvalidated vulnerabilities $q_a(t_n)$ and cases of validations that led to critical errors $q_c(t_n)$

Normalized time – t_n	**0**	**0,022**	**0,03**	**0,126**	**0,129**	**0,145**	**0,148**	**0,188**	**0,191**	**0,756**	**0,788**	**1**	**0**
$q_s(t_n)$	0	0	1	3	1	0	3	1	3	3	0	3	0
$q_f(t_n)$	0	32	40	58	58	64	53	82	60	1442	1255	1908	0
$q_c(t_n)$	0	0	0	2	0	1	2	1	2	0	0	0	0

*Note. Compiled based on statistical data of an experimental study of the functioning of the Metasploit-autopwn automated tool for exploiting vulnerabilities (**Table 4.1**)*

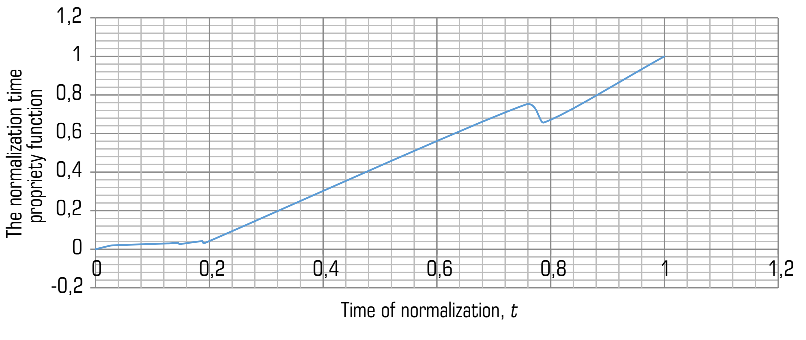


Fig. 4.11 Vulnerability validation analysis normalization time appropriateness function

Fig. 4.11 shows that the value of the membership function of the normalization time coincides with the error values E, which is determined by dependence (4.21). Thus, as the normalization time increases, the probability of an error in the implementation of specific exploits increases.

Each of the parameters determined by dependencies (4.20)–(4.22) has its own ranges of values independently of each other.

Therefore, there is a vague scale of the three values of these parameters, on the basis of which it is already possible to build a knowledge base for decision-making when validating corporate network vulnerabilities.

At the same time, a universal scale of linguistic variables (terms) was used for a vague scale of the quality of the vulnerability validation mechanism [80]:

$$T=\{Min, Low, Med, High, Max\}. \qquad (4.26)$$

Thus, based on the results of two independent analyzes of the validation process of identified vulnerabilities, vague evaluations of the quality of the vulnerability validation mechanism were formed and described in **Tables 4.12** and **4.13**.

Table 4.12 A knowledge base formed on the first experiment results using Armitage

A	***E***	***Ce***	Q_{mv}	**Description**
1	0	0	*Max*	A reliable vulnerability validation mechanism that does not disrupt the operation of target systems and does not allow wrong decisions regarding the use of exploits
[0,8;1)	(0:0,1)	(0:0,01]	*High*	The ability of the validation mechanism to successfully check and confirm the possibility of implementing vulnerabilities due to correctly made decisions regarding the use of selected exploits is high, with a fairly low number of wrong decisions
[0,5;0,8]	[0,1;0,35]	(0,01;0,2)	*Med*	The validation mechanism is quite stable
[0,1;0,5)	(0,35;0,7]	[0,2;0,5)	*Low*	For the most part, the vulnerability validation mechanism is inactive (inefficient) because it allows an unacceptable number of wrong decisions regarding the use of exploits
[0;0,1)	(0,7;1]	[0,5;1]	*Min*	The validation mechanism is unusable because it causes too many failures in the target systems

Table 4.13 A knowledge base formed on the second experiment results using Metasploit-autopwn

A	*E*	*Ce*	Q_{mv}	Description
(0,8;1]	[0;0,1)	[0;0,1)	*Max*	A reliable vulnerability validation mechanism that practically does not lead to disruption of target systems, and also allows a minimum number of wrong decisions regarding the use of exploit
(0,7;0,8]	[0,1;0,2)	[0,1;0,15)	*High*	The ability of the validation mechanism to successfully check and confirm the possibility of implementing vulnerabilities due to correctly made decisions regarding the use of selected exploits is quite high, with a low number of false decisions
[0,5;0,7]	[0,2;0,3]	[0,15;0,2]	*Med*	The validation mechanism is quite stable, however, it allows correct validation of vulnerabilities in no more than 70 % of cases
[0,1;0,5)	(0,3;0,7]	(0,2;0,4]	*Low*	For the most part, the vulnerability validation mechanism is ineffective (inefficient) because it allows an unacceptable number of wrong decisions regarding the use of exploits
[0;0,1)	(0,7;1]	(0,4;1]	*Min*	The validation mechanism is not recommended for use because it leads to a large number of failures in the functioning of the target systems

It can be seen from both tables that when conducting two independent experiments with obtaining a large volume of statistical data, unclear estimates were formed Q_{mv} of the performance quality of the vulnerability validation mechanism, which is based on the introduction of set terms (2.10), does not differ significantly.

Based on this, and also taking into account the expression (4.27), according to which it was established that the knowledge base should contain 125 rules of logical inference of the form (4.28) [85, 92, 93], a generalized knowledge base was built, a fragment of which is presented in the form of **Table 4.14**.

$$N_{max}=l_1 \cdot l_2 \cdot \ldots \cdot l_n \tag{4.27}$$

where N_{max} – the number of terms for the evaluation of the i_{th} output variable $\left(i=\overline{1,n}\right)$; n – the number of output variables.

$$R(i): IF\left(A_i \in T \,\&\, E_i \in T \,\&\, Ce_i \in T\right) THEN\left(Q_{mv_i} \in T\right),\ i=\overline{1,k}. \tag{4.28}$$

Table 4.14 Knowledge base fragment

№	*A*	*E*	*Ce*	Q_{mv}	№	*A*	*E*	*Ce*	Q_{mv}
1.	*Max*	*Max*	*Max*	*Max*	7.	*Max*	*High*	*High*	*High*
2.	*Max*	*Max*	*High*	*Max*	8.	*Max*	*High*	*Med*	*High*
3.	*Max*	*Max*	*Med*	*Max*	9.	*Max*	*High*	*Low*	*Med*
4.	*Max*	*Max*	*Low*	*High*	10.	*Max*	*High*	*Min*	*Med*
5.	*Max*	*Max*	*Min*	*Med*	…	…	…	…	…
6.	*Max*	*High*	*Max*	*Max*	125.	*Min*	*Min*	*Min*	*Min*

The built knowledge base made it possible to form decisive decision-making rules (**Table 4.15**) regarding the implementation of one or another attacking action, taking into account the quality rank of the vulnerability exploit.

Table 4.15 Decisive decision-making rules regarding the implementation of vulnerability exploits

Rank/ Q_{mv}	**Max**	**High**	**Med**	**Low**	**Min**
Excellent	a_1	a_1	a_1	a_1	a_1
Great	a_1	a_1	a_1	a_1	a_1
Good	a_1	a_1	a_1	a_1	a_1
Normal	a_1	a_1	a_1	a_2	a_2
Average	a_1	a_1	a_2	a_2	a_2
Low	a_1	a_2	a_2	a_2	a_2
Manual	a_2	a_2	a_2	a_2	a_2

where Rank – exploit quality rank; a_1 – implement the selected vulnerability exploit; a_2 – skip the selected vulnerability exploit.

In turn, the formed decisive rules allow developing expert systems [6] for automating the decision-making process when validating identified vulnerabilities of target information systems and networks.

4.5 A METHOD OF AUTOMATIC ACTIVE ANALYSIS OF THE CORPORATE NETWORKS SECURITY BASED ON VULNERABILITIES INTELLIGENT VALIDATION

The proposed method of automatic active analysis of the security of corporate networks defines the main stages of using the developed: mathematical model for the analysis of quantitative characteristics of the vulnerability validation process, methods for analyzing the quality of work of

the mechanism for validating detected vulnerabilities of the corporate network, and a method for building a fuzzy knowledge base for decision-making in the validation of software and hardware platform vulnerabilities. At the same time, the method can be divided into 4 main stages (**Fig. 4.12**): (I) preparatory stage, (II) initialization stage, (III) stage of adaptive validation of probable vulnerabilities, (IV) stage of processing and display of results (determination of the actual security level).

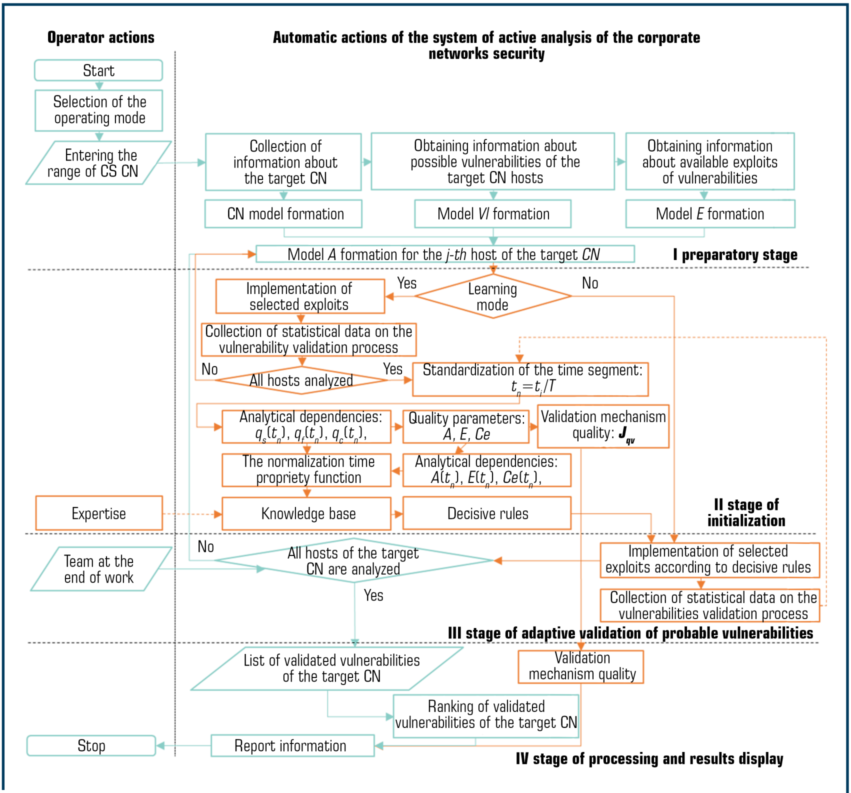


Fig. 4.12 Scheme of the method of automatic active analysis of the corporate network's security based on vulnerabilities intelligent validation

It should be noted that the proposed method includes two modes of operation, the first is training, during which all the above-mentioned scientific results are implemented for the construction and adaptation of the knowledge base, as well as decisive rules, that is, the training of the automatic system of active analysis of the security of corporate networks is carried out, and the second mode – directly the active analysis of the security of the corporate network.

In addition, on the basis of the analysis of approaches to conducting an active analysis of the security of corporate networks and methods and means of its automation, respectively, a number of models were formed that will allow the use of information about the target corporate network as input data, in particular information about all its components, found vulnerabilities and exploits available for their implementation.

The structure of these models is described using a theoretical-multiple approach.

Next, the sequence of steps of the method is considered in more detail:

Step 1. Gathering information about the target corporate network and forming a model *CN* according to (4.14). Any modern security scanner can be used as a source of all the necessary information, in particular information about the configuration of individual hosts of the target corporate network.

$$CN = \left\langle H, T_H, I_H \right\rangle, \tag{4.29}$$

where $H = \left\{ h_1, \ldots, h_j \right\}$ – finite set (f.s.) of hosts (nodes) of the corporate network; T_H – the type of the j_{th} host; I_H – key information about the target j_{th} host.

The host type is represented as [114]:

$$T_H = \left\{ CS, NH, M \right\}, \tag{4.30}$$

where CS – computer system; NH – network equipment; M – mobile platform.

Information about the target host:

$$I_H = \left\{ Pl, V_{Pl}, S, V_S, P \right\}, \tag{4.31}$$

where $Pl = \left\{ pl_1, \ldots, pl_j \right\}$ – f.s. of platforms (Windows, Linux, Android and other); $V_{Pl} = \left\{ v_{pl_1}, \ldots, v_{pl_j} \right\}$ – f.s. of probable platform versions; $S = \left\{ s_1, \ldots, s_j \right\}$ – f.s. of services; $V_S = \left\{ v_{s_1}, \ldots, v_{s_j} \right\}$ – f.s. of probable names and versions of the relevant services; $P = \left\{ p_1, \ldots, p_j \right\}$ – f.s. of ports on which services are running and running.

It should be noted that this description is built according to the Common Platform Enumeration (CPE) standard, which allows later, when making an assumption about the presence of vulnerabilities in the target system, to link the host configuration with data from the vulnerability database and to select appropriate exploits.

As an example: $\left\langle Linux,\ 2.6.x,\ ftp,\ \mathrm{Pr}\,oFTPD\ 1.3.1.2121 \right\rangle$.

Step 2. Obtaining information about possible vulnerabilities of the hosts of the target corporate network and forming a Vl model according to (4.32). The main source of information about vulnerabilities is open databases of vulnerabilities.

$$Vl = \left\langle ID_{Vl}, R_{Vl}, C_{Vl} \right\rangle, \tag{4.32}$$

where $ID_{Vl} = \left\{ id_{vl_1},...,id_{vl_n} \right\}$ – f.s. of identifiers of vulnerabilities presented in the CVE List; $R_{Vl} = \left\{ r_{vl_1},...,r_{vl_n} \right\}$ – f.s. of criticality assessments of vulnerabilities according to CVSS; $C_{Vl} = \left\{ c_{vl_1},...,c_{vl_n} \right\}$ – f.s. of known vulnerable configurations (identifiers issued using Common Platform Enumeration).

Step 3. Obtaining information about available exploits and forming a model E according to (4.33). Accordingly, the source of the necessary information is open and closed databases of exploits, ready-made exploit kits or integrated databases directly of the exploitation tools themselves.

$$E = \left\langle N_E, D_E, R_E, Rf_E \right\rangle, \tag{4.33}$$

where $N_E = \left\{ n_{E_1},...,n_{E_g} \right\}$ – f.s. of short names of available exploits (in fact, identifiers are represented by one or another means of exploitation); $D_E = \left\{ d_{E_1},...,d_{E_g} \right\}$ – f.s. of short descriptions of exploits (in which the name and version of the vulnerable service are indicated); $R_E = \left\{ excellent,...,manual \right\}$ – f.s. of exploit quality ranks, $Rf_E = \left\{ rf_{E_1},...,rf_{E_g} \right\}$ – f.s. of links to identifiers of vulnerabilities that are implemented using an exploit.

Step 4. Selection of exploits of vulnerabilities for the jth host of the target network according to the cyber attack model A (using vulnerabilities), which is formed based on compliance with the main characteristics and vulnerabilities of the target system:

$$A = \left\{ a_1,...,a_k \right\}, \tag{4.34}$$

where $a_k = F\Big(\big(\left(Vl.C_{Vl}, I_H.S, I_H.V_S \right) \& \left(E.Rf_E, Vl.ID_{Vl} \right) \big) \Big| \left(E.D_E, I_H.S, I_H.V_S \right) \Big)$.

Step 5 (in learning mode). The implementation of selected exploits and the collection of statistical data on the basic characteristics of the target host vulnerability validation process are carried out one by one. Based on the collected data, the quality of the vulnerability validation mechanism is evaluated according to the following sub-steps:

5.1. Standardization of the time segment for validation of host vulnerabilities of the target corporate network according to (4.8):

$$t_n = t_i / T;$$

5.2. Obtaining analytical dependencies for basic characteristics $\left(q_s, q_f, q_c \right)$ of vulnerability validation process (4.30):

$$q_s(t_n) = \sum_{i=0}^{n} q_s(t_n^{(i)}) b_{k,n}(t_n),\ q_f(t_n) = \sum_{i=0}^{n} q_f(t_n^{(i)}) b_{k,n}(t_n),\ q_c(t_n) = \sum_{i=0}^{n} q_c(t_n^{(i)}) b_{k,n}(t_n).$$

5.3. Calculation of quality indicators of the vulnerability validation mechanism (4.20)–(4.22): A – accuracy, E – error and Ce – critical error:

$$A=\frac{\int_0^1 q_s(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}\ ; \quad E=\frac{\int_0^1 q_f(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}\ ;$$

$$Ce=\frac{\int_0^1 q_c(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}.$$

5.4. Evaluation of the quality of the validation mechanism according to the single integral quality indicator (4.23):

$$J_{qv}=\frac{A}{E}-Ce.$$

5.5. Obtaining analytical dependencies for the quality indicators of the vulnerability validation mechanism $A(t_n)$, $E(t_n)$, $Ce(t_n)$ (4.28)–(4.30):

$$A(t_n)=\frac{\int_0^{t_n} q_s(\theta)d\theta}{\int_0^{t_n} (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}; \quad E(t_n)=\frac{\int_0^{t_n} q_f(\theta)d\theta}{\int_0^{t_n} (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta};$$

$$Ce(t_n)=\frac{\int_0^{t_n} q_c(\theta)d\theta}{\int_0^{t_n} (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}.$$

5.6. Construction of the normalization time membership function.

5.7. Formation of the knowledge base and decisive decision-making rules regarding the implementation of the attacking action in the form of (4.28) logical conclusion

$$R(i): IF\left(A_i \in T\ \&\ E_i \in T\ \&\ Ce_i \in T\right) THEN\left(Q_{mv_i} \in T\right),\ i=\overline{1,k},$$

having previously determined the required number of rules according to (4.27): $N_{\max}=l_1\cdot l_2\cdot\ldots\cdot l_n$.

Step 5 (in active analysis mode). Implementation of selected exploits in accordance with decisive decision-making rules and collection of statistical data regarding the vulnerability validation process, based on which, in accordance with Steps 5.1–5.4, the quality assessment of the vulnerability validation mechanism is carried out.

Step 6 (in active analysis mode). Ranking of validated vulnerabilities of the target corporate network and generating a report of the conducted active security analysis. After analyzing all the hosts of the target corporate network, a general list of validated, i.e., confirmed vulnerabilities is formed, at the same time, they are ranked according to the level of their criticality, which is determined by the CVSS base assessment and the level of prevalence of this vulnerability L_{v_i} in the corporate network according to the expression (4.35), otherwise, there is a return to Step 5 (in active analysis mode). As a result, a report of the conducted active security analysis is generated, containing a ranked list of confirmed vulnerabilities of the target corporate network in descending order, from vulnerabilities with the highest levels of criticality and prevalence to vulnerabilities with the lowest levels, as well as the quality level of the mechanism for validating the identified vulnerabilities.

$$L_{v_i} = \frac{h_v}{h_T} \cdot 100, \tag{4.35}$$

where h_v – number of vulnerable hosts to the validated vulnerability *v*; h_T – the total number of analyzed hosts of the target corporate network, $h_T > 0$.

Based on the report, the expert decides on the necessity of retraining the automated system of active security analysis, as well as on the priority elimination of one or another validated vulnerability.

5 CRYPTOSYSTEMS BASED ON CELLULAR AUTOMATA

ABSTRACT

The chapter is deal with the development of cryptographic primitives based on cellular automata. The definition of cellular automata is given and the elementary rules of intercellular interaction are described.

A number of generators of pseudorandom binary sequences have been developed based on a combination of elementary rules of intercellular interaction, as well as cell interaction according to a rule of our own development.

In the "cryptographic sponge" architecture, a cryptographic hashing function with a shuffling function based on cellular automata was developed and its statistical characteristics and avalanche effect were investigated.

A block cipher in the SP-network architecture is constructed, in which cellular automata are used to deploy the key, and the encryption process is based on elementary procedures of replacement and permutation. Substitution blocks are used from the well-known AES cipher.

The last section is deal with the description of a stream cipher, where the keyboard and mouse of a personal computer are used as the initial entropy. Random data received from the specified devices is processed by a proprietary hashing function based on a "cryptographic sponge".

All developed cryptographic functions and primitives demonstrate good statistical characteristics and avalanche properties.

KEYWORDS

Cellular automata, pseudorandom binary sequences, cryptographic hashing function, block cipher, stream cipher.

5.1 THE CONCEPT OF CELLULAR AUTOMATA AND THEIR APPLICATION

Cryptographic protection plays a leading role in this case. In this regard, any research and development of scientists related to the improvement of cryptographic protection is considered relevant.

On the other hand, for the construction of cryptographic primitives of various kinds, cellular automata (CA) are widely used [117–120]. For the first time, this possibility was noted by S. Wolfram [121]. Since then, a number of cryptographic transformations have been developed on the basis of CA: symmetric ciphers, hashing functions, etc. [117, 118, 122].

Nevertheless, the topic of designing cryptosystems based on CA continues to develop rapidly, since the simplicity of the architecture and the possibility of multi-threaded implementation allow improving the statistical and cryptographic characteristics of such cryptosystems.

This chapter is devoted to the development of two symmetric ciphers based on CA: block and stream ciphers, which can be used to protect both individual messages, files, and communication channels.

A cellular automata is a discrete mathematical model that defines a population and is described by a set of cells that form a periodic grid and specified transition rules that determine the state of the cell based on the current state of the cell itself and those of its neighbors located at a certain distance from it, which does not exceed maximum [121].

The main direction of cellular automata research is algorithmic solvability of individual problems. The issues of constructing the initial states in which the cellular automata will solve the given problem are considered too.

Classical CA generally meet the following criteria:

– changing the values of all cells occur simultaneously after calculating the new state of each grid cell. Otherwise, the order of selecting grid cells during the iterative process would significantly affect the result;

– the grid is uniform. It is impossible to distinguish any two locations on the grid across the landscape. However, in practice, the grid turns out to be a finite set of cells (because it is impossible to allocate an unlimited amount of data). As a result, edge effects may occur: cells standing on the borders of the lattice will differ in the number of neighbors. To avoid this, it is possible to introduce periodic boundary conditions (for example, close the grid, then, for example, the first and last, upper and lower elements will be neighbors);

– interactions are local. Only surrounding cells (as a rule, neighboring cells) can affect this cell;

– the set of cell states is finite. This condition is necessary so that a finite number of operations must be performed to obtain a new cell state value.

All cellular automata have some properties common to all.

States of elements. At each moment of time, each element of the CA takes one state from a finite set of states. Depending on these states, at the next moment in time, the set of elements can take a new state. In practice, cells with an algebraic structure equivalent to sets of possible states are used – linear CAs.

Geometry. Elements can be geometrically arranged in various ways. The dimensionality of the space can be arbitrary, and the number of elements – both infinite and finite. In dynamic CA, the geometry can change over time, and if the geometry is different in different parts of space, such cellular ones are called heterogeneous.

Neighborhood. Neighbors are elements on which the CA element depends. The concept of neighborhood can be called key for CA. The state of the element at the next moment in time is calculated from the state of the element itself and its neighbors. The neighborhood is determined to a greater extent by the geometry of the CA (when designing the CA, the desired number of neighbors

is chosen, which depends on the state of the element and their placement in relation to the element). For different purposes, it is possible to change the number of input states of the element.

Local rule. According to the local rule, the state of the CA element changes over time. CA, in which the local rules are different for different elements, is called heterogeneous. A local rule can be non-deterministic, that is, change over time or have a random nature.

An elementary CA is the simplest possible variant of a CA, it works in a one-dimensional space (line), its cells can have only two states (0 or 1), and the rule for determining the next state of a cell is determined only by its current state and the state of its two nearest neighbors.

S. Wolfram [121] proposed the scheme for representing the rules of elementary CA in the form of binary numbers from 0 to 255. Each possible configuration of the current cell and its neighbors is written in the following order: 111, 110, 101, 100, 011, 010, 001, 000. Then, under each configuration, the next state of the cell is written, as a result, the 8 received bits of the new state form a decimal number from 0 up to 255 in binary representation. This number will be the CA rule.

For a visual demonstration of the behavior of such CA, it is customary to record its generations from top to bottom. Basically, the behavior is investigated with a single "living" cell in the center (its state is 1, and the state of all the rest is 0), or the initial generation is set randomly [121]. **Fig. 5.1** shows an example of the graphic output of an elementary CA according to the rule of 90. If to read the state of the automata from right to left, let's get 01011010, which is the number 90 in the decimal number system.

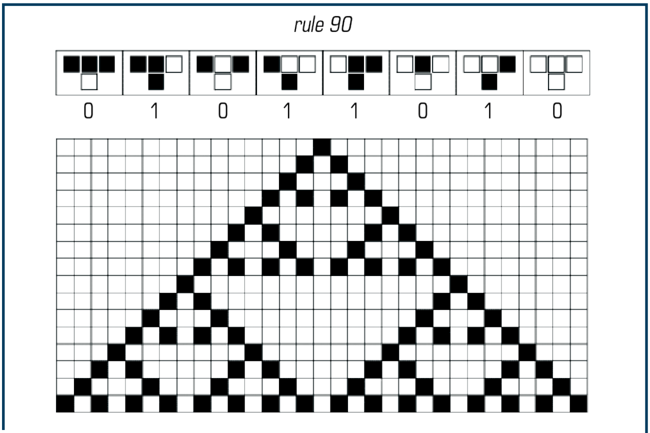


Fig. 5.1 Sierpinski napkin for rule 90

The lower part of the figure demonstrates the so-called "Sierpinski napkin", which is formed, as a rule, from sixteen generations of CA, the initial state of which consists of a central "one", while the rest of the cells are logical "zeros". Next, the process of interaction is started according to a

certain rule (here – rule 90), and the result is displayed in the figure with dark and light cells. After 16 stages of interaction, there is the picture presented in the figure. Of course, cellular automata are not only one-dimensional. Two- and three-dimensional CAs are used for simulation modeling of complex mathematical, physical, biological, social and other structures (for example [122–126]).

Let's consider some applications of the cellular-automatic method of simulation modeling.

Examples of simulation modeling by cell-automata methods

For an example of the possibilities of simulation modeling by the method of asynchronous cellular automata, consider the model of some system of interacting objects, which is an improvement of the classic "predator-prey" system [122]. The essence of the modification is to increase the interaction functionality. The "victim" can be water in the soil, and the "predators" will be some entities for the "existence" of which water consumption is vital. Let's consider the increase in the water level to be an arbitrary random process. Let's also consider the possibility of gradient movement of water in the soil. Water consumers, in turn, must have two main functions: the ability to move in a chaotic manner and the ability to burrow into the soil. Let the soil also have the property of self-leveling, that is, the deeper the water user goes into the soil, the more intensively it is pushed to the surface.

Let's analyze the described model. With the complete absence of consumers and an arbitrary level of groundwater, the latter will rise to the surface over time. With the appearance of consumers in case of water saturation, the process of intensive consumption and, accordingly, intensive growth of the number of consumers will begin. The number of dead consumers will also increase intensively, since the process of "competition" will be involved, when all the individuals that were pushed to the surface will not "survive" because there is not enough nutrient medium. If at the initial stage of modeling only the movement of water consumers is predicted, then the dynamics of the system will be wave-like (self-oscillating). The process of competition leads to the fact that the system acquires a stationary character as a cellular structure, as shown in **Fig. 5.2**. The interaction functions are given by system (5.1).

$$\begin{cases} c_1^i = c_1^i + k_{11} \cdot c_1^i + k_{12} \cdot s \cdot \left(c_1^2 - c_1^1\right) + k_{13} \cdot c_1^i \cdot c_4^i \cdot \left(c_1^{=} - c_1^i\right) \times \\ \times \left[1 - 1 / \left(1 + \exp\left(10 \cdot \left(c_1^i - c_1^{thr}\right)\right)\right)\right] \\ c_2^i = c_2^i + k_{21} \cdot \left(c_2^s - c_2^i\right) + k_{22} \cdot s \cdot \left(c_2^2 - c_2^1\right) + k_{23} \cdot c_1^i \cdot c_2^i + k_{24} \cdot c_2^i \cdot \left|c_4^2 + c_2^2 - c_4^1 - c_2^1\right| \\ c_3^i = c_3^i + k_{31} \cdot \left(c_3^s - c_3^i\right) + k_{32} \cdot s \cdot \left(c_3^2 - c_3^1\right) + k_{33} \cdot \left(c_2^i - c_3^i\right) / \\ / \left(1 + \exp\left(10 \cdot \left(c_2^i - c_3^i\right)\right)\right) + k_{34} \cdot c_3^i \cdot \left|c_4^2 + c_2^2 - c_4^1 - c_2^1\right| \\ c_4^i = c_4^i + k_{41} \cdot s \cdot \left(c_4^2 + c_2^2 - c_4^1 - c_2^1\right) + k_{42} \cdot c_4^i \cdot c_1^i + k_{43} \times \left(c_2^i - c_3^i\right) / \\ / \left(1 + \exp\left(10 \cdot \left(c_2^i - c_3^i\right)\right)\right) \end{cases}, \qquad (5.1)$$

where $i = 1,2$ – indexes of interacting cells; $s = +1$ at $i = 1$, $s = -1$ at $i = 2$. System (1) describes the 4-layer structure of the two-dimensional field of cellular automata. Layer c_1^i corresponds to the concentration of water consumers; c_2^i – ground level, c_3^i – groundwater level, c_4^i – surface water level, k_{ij} – activity coefficients that acquire the following values (for the case shown in **Fig. 5.2**): $k_{11} = -0.1$ – intensity of the arbitrary decrease in the concentration of consumers, k_{12} – intensity of gradient displacement of consumers; $k_{13} = 0.0001$ – intensity of the increase in consumers due to the absorption of surface water ($c_1^s = 100$ – maximum concentration (saturation), $A_1^{thr} = 10$ – threshold value for the concentration of consumers, when intensive growth in the process of water absorption begins, $k_{21} = 0.005$ – intensity of soil level rise ($c_2^s = 100$ – maximum concentration (saturation)), $k_{22} = 0.3$ – intensity of the gradient displacement of the soil, k_{23} – intensity of soil digging by consumers, $k_{24} = -0.01$ – the intensity of soil erosion with the flow of groundwater, $k_{31} = 0.01$ – intensity of the groundwater level rise ($c_2^s = 100$ – maximum concentration (saturation)), $k_{32} = 0.5$ – intensity of gradient displacement of groundwater, $k_{33} = 1$ – intensity of the decrease in the level of groundwater due to its transformation into surface water, $k_{34} = -0.01$ – intensity of the decrease in the groundwater level due to soil erosion during the displacement of the groundwater level, $k_{41} = 0.5$ – intensity of the gradient movement of surface water across the soil surface, $k_{42} = -0.005$ – intensity of the decrease in the level of surface water due to its absorption by consumers, $k_{43} = -1$ – intensity of the increase in the level of surface water due to its transformation from ground water.

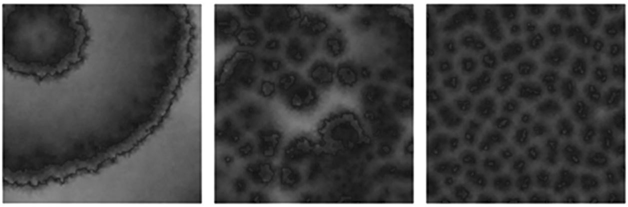


Fig. 5.2 Evolution of a wave-like structure to a system with a cellular structure

Activity coefficients k_{12} (intensity of gradient movement of consumers) and k_{23} (the intensity of soil digging by consumers) change randomly, and this change may obey some laws of "labor activity": the particle either moves across the field, or it is buried in the soil. Denote by ξ a random variable that varies randomly in the range [0...1], then the activity coefficients k_{12} and k_{23} served as follows: $k_{12} = 0.5 \cdot \xi$, $k_{23} = 0.01 \cdot (1 - \xi)$. In addition, let's relate the degree of chaoticity ξ with the population level of the field cell, i.e. if the concentration of consumers decreases, then the intensity of the chaotic change ξ acquires non-zero values (for example, proportional to the ratio of the difference in the concentrations of consumers before and after the interaction to the max-

imum possible concentration of consumers in two cells), but with an increase in the concentration of consumers, the intensity of chaotic change ξ is equal to zero. Thus, the value ξ the field of an individual cell can change, which leads to competition of cell properties, and, as can be seen from **Fig. 5.2**, the most "survival" are those cells for which $\xi \to 0$.

System (1) also takes into account soil erosion during the movement of surface water. This process can lead to the emergence of a branch-like structure of the soil relief. Since surface water moves along the surface gradient and at the same time erodes it, the accidental formation of a depression will increase in size due to the following mechanisms: increased water flows due to the gradient increase of surface water, erosion of the soil by the water flow, self-leveling of the soil. Adding such mechanisms to the model leads to the result shown in **Fig. 5.3**.

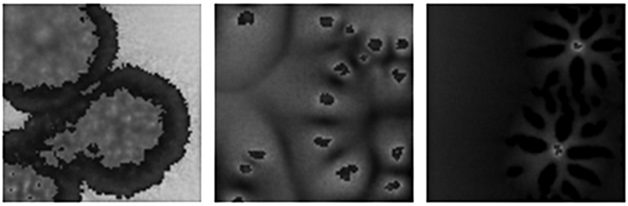


Fig. 5.3 Evolution of a wave-like structure to a branch-like system

The interaction functions in this case are also given by system (1), but the values of the activity coefficients are slightly different: $k_{11} = -0.3$, $k_{12} = 0.5 \cdot \xi$, $k_{13} = 0.0001$, $k_{21} = 0.005$, $k_{22} = 0.03$, $k_{23} = 0.01 \cdot (1 - \xi)$, $k_{24} = -0.007$, $k_{31} = 0.015$, $k_{32} = 0.05$, $k_{33} = 1$, $k_{34} = -0.007$, $k_{41} = 0.5$, $k_{42} = -0.0003$, $k_{43} = -1$. As can be seen from **Fig. 5.3**, in the process of consumer competition, those whose main property is burial in the soil also survive, but those entities around which a branch-like structure of nutrient inflow has formed have an advantage.

The given example, although it does not provide a complete understanding of the capabilities of the cellular-automata approach to simulation modeling, however, demonstrates its flexibility and the possibility of application to almost any system. Mathematical, physical, biological, and social phenomena are conveniently and effectively modeled by such methods, which makes them an indispensable and useful tool for simulation modeling. In the future, let's focus on cryptographic applications of the cellular automaton approach.

5.2 CELLULAR AUTOMATA IN CRYPTOGRAPHIC TRANSFORMATIONS

Cellular automata have long been used to construct cryptographic primitives [120, 126–128]. Often, the elementary rules of one-dimensional CAs are used to generate binary pseudorandom

sequences that have good statistical characteristics (for example [128]). It is also known to use CA for building hashing functions [117, 118], stream and block ciphers [120, 127, 128], etc. In our research, let's use the following elementary rules of one-dimensional CAs, which are presented in **Table 5.1**.

Table 5.1 Researched rules of intercellular interaction of CA

No	Rule	Boolean form	Arithmetic form
1	«22»	$b' = a \oplus a \wedge b \wedge c \oplus b \oplus c$	$b' = ((a + b + c + abc)\text{mod}2)$
2	«30»	$b' = a \oplus (b \vee c)$	$b' = ((a + b + c + bc)\text{mod}2)$
3	«54»	$b' = (a \vee c) \oplus b$	$b' = ((a + b + c + bc)\text{mod}2)$
4	«86»	$b' = (a \vee b) \oplus c$	$b' = ((a + b + ab + c)\text{mod}2)$
5	«105»	$b' = \overline{(a \oplus b \oplus c)}$	$b' = ((1 + a + b + c)\text{mod}2)$
6	«135»	$b' = 1 \vee a \oplus b \vee c$	$b' = ((1 + a + bc)\text{mod}2)$
7	«146»	$b' = (a \vee c) \wedge (a \vee b \vee c)$	$b' = ((a + ab + c + bc + abc)\text{mod}2)$
8	«149»	$b' = a \vee b \oplus c \vee 1$	$b' = ((1 + ab + c)\text{mod}2)$
9	«150»	$b' = a \oplus b \oplus c$	$b' = ((a + b + c)\text{mod}2)$
10	«158»	$b' = a \oplus b \oplus c \vee b \wedge c$	$b' = ((a + b + c + bc + abc)\text{mod}2)$

Rules "30", "86", "135" and "149" and their combinations are most suitable for building generators of binary pseudorandom sequences, others rules – for developing round transformations of cryptographic hash functions and block ciphers.

This does not mean that other rules of intercellular interaction (and there are 256 of them in total according to S. Wolfram [8]) not given in this table, are unsuitable for cryptographic transformations; however, the authors used exactly these rules for their designs.

Pseudorandom sequences generators

As already mentioned, have developed a number of original generators of pseudorandom binary sequences that can be used in cryptographic systems based on cellular automata.

Based on the considered rules of intercellular interaction, a number of generators of binary sequences were developed and their statistical characteristics were investigated. The research was performed using the NIST STS v.1.8 statistical research package.

Fig. 5.4 presents statistical portraits of generators based on rules "30" and "22".

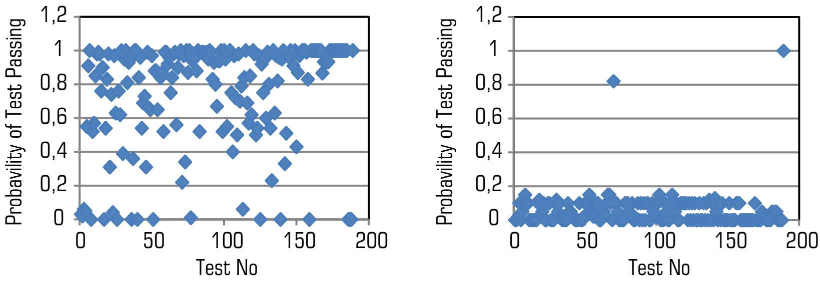


Fig. 5.4 Statistical portraits of generators based on rules "30" (left) and "22" (right)

As can be seen from the figure, the portraits of the generators are significantly different. The generator based on the rule of "22" did not pass almost any NIST STS test, so its use for developing cryptographically stable applications is not promising. Similar results were obtained for rules "54", "150" and "178". This cannot be said about rules "30", "86", "135" and "149", the statistics of which suggest that they can be successfully used, subject to appropriate modification, for the development of cryptographically stable generators of pseudorandom binary sequences.

Now it is necessary to investigate the effect of modifications of the selected rules on their statistical characteristics. As a modification, an additional combination of array cells was chosen for output, since the output of any one bit does not lead to the desired sequence quality.

Several options are possible here: combining based on simple logical operations; pseudo-random cell selection for output, as well as modification of the very rules of intercellular interaction. It would also be interesting to try alternative rules of intercellular interaction that do not belong to elementary ones.

In the first case, let's derive the result of addition modulo two of the contained three cells:

$$M_{out} = M[w] \oplus M[v] \oplus M[t]; \tag{5.2}$$

where w, v, t are cell numbers of the automata. In our case, cell numbers were taken as constant. It is also possible to select cell numbers by key.

In the second case, let's use the following formula to calculate the output cell number:

$$w = \left\{ i + 3 + \sum_{k=0}^{\log_2(n-1)} c_{i+3+k} 2^k \right\} \bmod n. \tag{5.3}$$

Here c is the content of the $(i+3+k)$-cell, n is the total number of cells in the machine; w is the number of the output cell.

Statistical portraits of generators with such modifications are presented in **Fig. 5.5**.

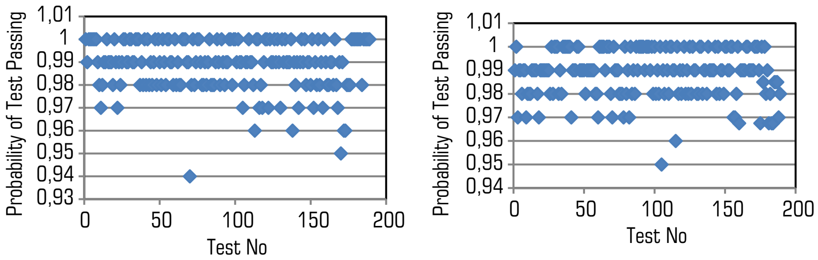


Fig. 5.5 Results of statistical testing of the generator based on the "30" rule with a combined output. On the left – according to formula (2), on the right – according to formula (3)

Similar results were obtained for the rules of intercellular interaction "86", "135" and "149" (**Fig. 5.6**).

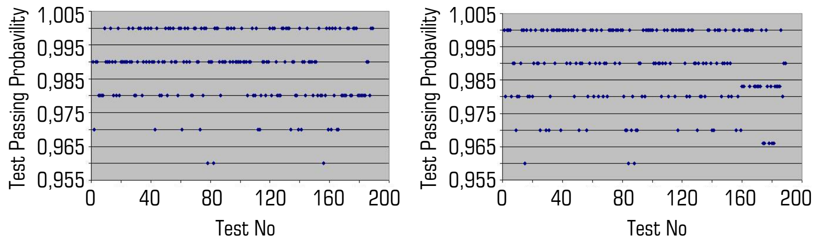


Fig. 5.6 Statistical portraits of generators based on rules "86" (left) and "149" (right)

The modification of the rules was reduced to the fact that not only the nearest neighbors were involved in the interaction, as provided by the elementary rules, but also arbitrary cells of the automaton. For example, rule "30" with elements of "far action" could look like this:

$$a'[i] = \{(a[i-1] + a[i] + a[i+k] + a[i]a[i+t]) \bmod 2\},$$

where i, $i+k$, $i+t$ are the numbers of interacting cells. The statistical portrait of such a generator with $k = 17$, $t = 23$ is presented in **Fig. 5.7**.

In all cases, the starting state was taken as the one obtained after 250 empty cycles of the generator from the position when the 128th bit was set to "1" and the rest of the 256 bits were set to "0". Let's consider here two of the most successful, in our opinion, applications, namely, a generator based on a combination of elementary rules of interaction and based on a custom rule of interaction.

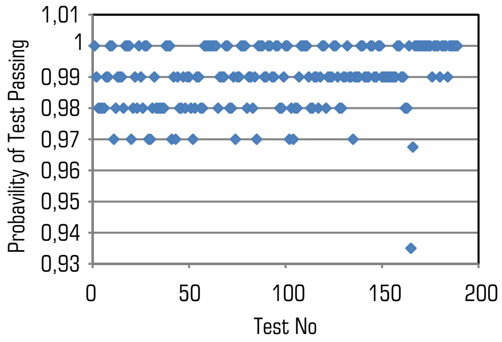


Fig. 5.7 Statistical portrait of the generator based on the "30" rule with "far echo"

Generator based on combined rules of intercellular interaction designed to maximize the statistical characteristics of the generated sequences.

The rules from one group according to S. Wolfram, which demonstrated the best statistical results, were chosen for research: this is the group of "rules 30". This includes rules "30", "86", "135" and "149".

The rules were combined as follows:

$$R_{out}[i] = \left(R_{30}[i] \oplus R_{86}[i] \oplus 1\right) \oplus \left(R_{30}[i] \oplus R_{135}[i] \oplus 1\right) \oplus \left(R_{30}[i] \oplus R_{149}[i] \oplus 1\right),$$

where $R_{out}\,[i]$ is the resulting transformation for cell number i; $R_{30}\,[i]$, $R_{86}\,[i]$, $R_{135}\,[i]$ and $R_{149}\,[i]$ – transformation of the i-th cell according to the rules "30", "86", "135" and "149", respectively.

It is clear that in this way we will sacrifice speed, since several interactions are applied to each cell, but for some cryptographic problems, the maximum cryptographic stability of the generated sequence is more important than the speed of the generator.

Another original result is a much faster pseudorandom sequence generator based on a proprietary cell-to-cell interaction rule. The gist is as follows. An additional array of addresses $A[i]$ was used to determine the numbers of interacting cells. This array contains decimal numbers of cells (their addresses), which are chosen pseudorandomly and duplicated in the buffer array $B[i]$. Selected cells interact according to a simple rule based on XOR:

$$C'_{A[i]} = C_{A[a]} \oplus C_{A[i+1]} \oplus C_{A[i]},$$

where $C'_{A[i]}$ is the resulting content of the cell numbered i; $C_{A[a]}$, $C_{A[i+1]}$, $C_{A[i]}$ are the contents of the corresponding cells; cell number a is determined as follows:

$$a = \left(i + 1 + \sum_{k=0}^{(\log_2 n)-1} C_{i+k+1} \times 2^{(\log_2 n)-k-1} \right) \bmod n.$$

Here n is the dimension of the cellular automata, that is, the number of cells in it.

After the interaction of i-th and a-th, the cells in the arrays change places. The process continues as long as necessary for the cryptographic application.

Graphically, the functioning of the proposed rule of intercellular interaction is presented in the figure (**Fig. 5.8**).

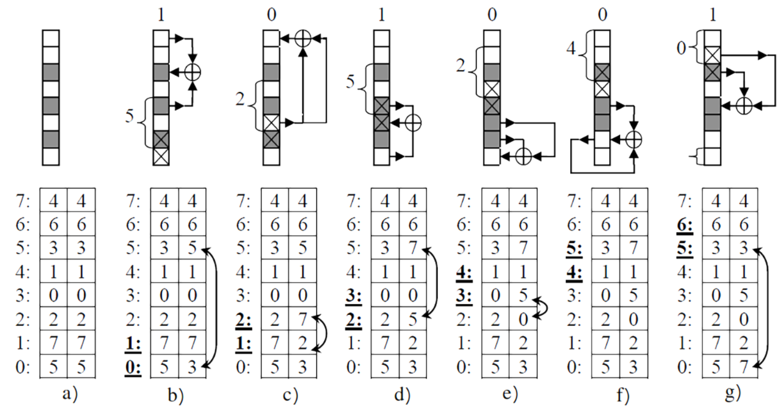


Fig. 5.8 Illustration of the proposed intercellular interaction algorithm

The upper part of the figure shows the successive states of the array of cellular automata; the lower one is an array of cell addresses (or indexes). Dark cells are logical ones, light cells are logical zeros.

In **Fig. 5.8 a**, the initial filling of the arrays is presented randomly; the CA array is filled with logical "1" and "0", the address array is filled with decimal numbers from 0 to 7. The address pointer is set to the beginning of the address array.

As can be understood from the figure, the algorithm works as follows.

Step 1 (**Fig. 5.8 b**). The address pointer is at the beginning of the address array (the null element is highlighted in bold and underlined). This element (the number "5") gives the address of the cell where the result of the interaction of the cells will be written (that is, the result will be written in the fifth cell).

Step 2. Addresses of interacting cells are determined as follows: the address of the first cell is contained in the element of the address array, the number of which is "pointer+1", that is, in our case, in the first element (its mark is also highlighted in bold and underlined). This element is

equal to 7, which means that the seventh cell of our spacecraft will participate in the interaction. The address of the second interacting cell is already calculated based on the binary values of the cells of the CA itself. It is possible to see that the zero and first cells in **Fig. 5.8 b**) are marked with crosses. The cross on the zero cell corresponds to the position of the pointer in the array of addresses, the cross on the first cell is the beginning of the calculation of the binary address. So there is $101_{(2)} = 5_{(10)}$. The fifth element of the address array is equal to 3, so the 7th and 3rd cells will interact, and the result is recorded in the 5th cell. This is shown in **Fig. 5.8 b**) by arrows.

Step 3. To improve the scattering effect, after the interaction of the CA cells, the element of the address array on which the pointer is set and the element whose number was determined by the binary method are swapped. In our case, these are the zero and fifth elements of the address array, which is indicated by the arrow in **Fig. 5.8 b**).

Step 4. The address array pointer moves to the next element.

Step 5. Steps 1–3 are repeated until the pointer reaches the end of the address array, after which the left and right columns of the address array are swapped, the pointer is again set to the zero element, and the process is repeated from the beginning.

Fig. 5.8 c–g shows several stages of the algorithm (from the first to the fifth element).

Statistical studies of the developed generators were carried out by the standard method using the NIST STS statistical test package. The obtained statistical portraits of the developed generators are presented in the **Fig. 5.9**.

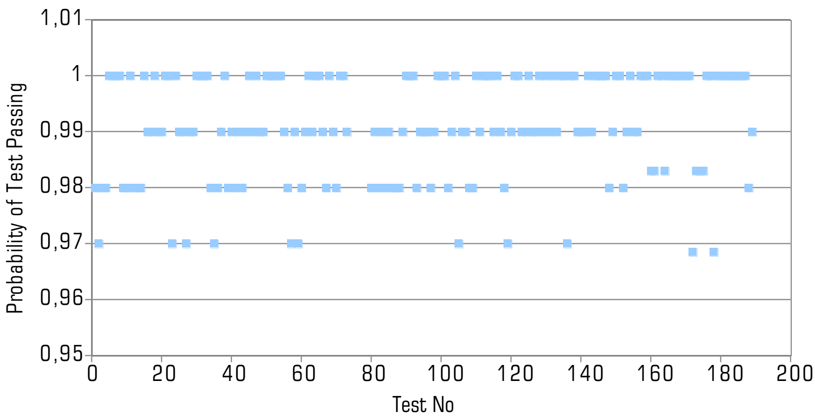


Fig. 5.9 Statistical portrait of a generator with a combination of intercellular interaction rules

Since it is assumed that the general sequence passed all tests, if at least 96 out of 100 subsequences passed it (which corresponds to the probability of 0.96 in the graph), it can be

seen that the generator based on the combination of elementary rules of interaction meets all the requirements for cryptographically stable pseudorandom sequences with a margin, so as the minimum probability values here are about 0.97.

Fig. 5.10 shows a statistical portrait of the generator based on its own interaction rule.

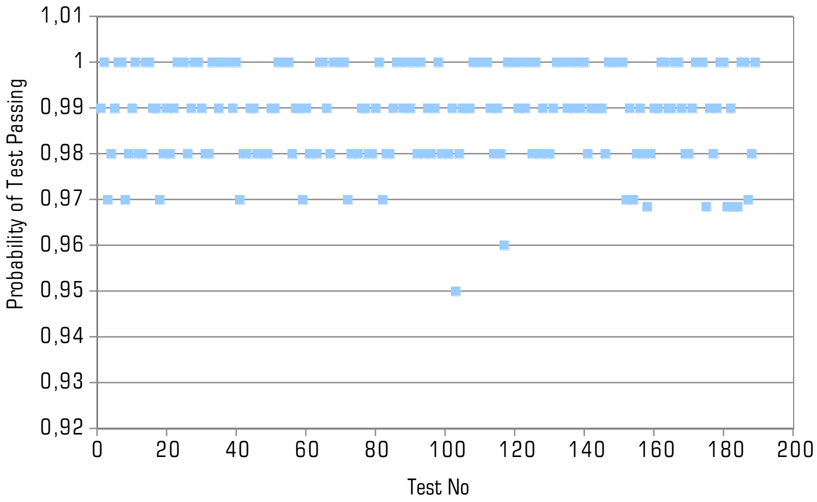


Fig. 5.10 Statistical portrait of a generator of pseudorandom sequences based on its own rule of intercellular interaction

It can be seen from the figure that this generator, in general, satisfies the requirements for cryptographically stable primitives, but there are two tests here that it passed at a level less than 0.96. This is not a major problem, so it is possible to recommend it for non-critical cryptographic applications with high-speed requirements.

Next, let's consider the hashing function and block cipher on a three-dimensional CA that uses simple rules of intercellular interaction.

5.3 HASHING FUNCTION BASED ON CELLULAR AUTOMATA

The Keccak hash function (SHA-3) was developed by J. Dyman and collaborators, the well-known developer of the ISO-standardized symmetric AES cipher.

Keccak is built on the so-called "cryptographic sponge" architecture [128, 129], the structure of which is shown in **Fig. 5.11**.

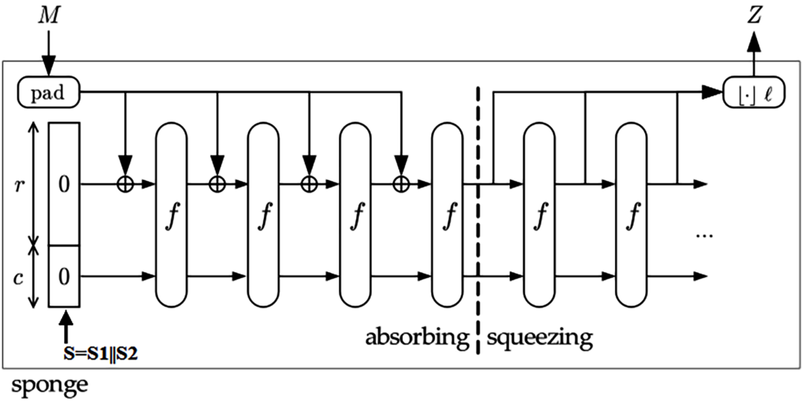


Fig. 5.11 Architecture of the "cryptographic sponge"

The main property of this architectural solution is that the input block of the open message (the length of which is much smaller than the size of the register $r||c$ – 1600 bits) is added to the initial state ($r||c$), which is processed by the shuffling function f. This function "blurs" the statistical characteristics of the incoming message to the entire internal state. The operation continues until all incoming messages are exhausted. Since compression is not used here, this process, according to the authors of the hashing function, is collision-free.

This mode of operation was called "absorbing" because the register seems to absorb the open message into itself. This is actually where the name of the architecture "cryptographic sponge" comes from.

The next stage in forming a hash image is the "squeezing" stage. It consists in the fact that portions of the hash image of 64 bits are extracted from the register containing the processed input message and, possibly, the initialization vector, from which the result is concatenated. In this case, the register is processed in the same way by the shuffling function. This procedure of forming the resulting image leads to the fact that it is practically impossible to establish a statistical relationship between the open message and the hash image.

The mixing function can be any. The mandatory condition is that it should make it as difficult as possible to find the connection between the incoming message and its hashing image.

We have made an attempt to develop a mixing function based on cellular automata and perform initial research of such a structure.

The developed mixing function is quite simple, takes into account the features of cellular automata and consists of the following steps.

Step 1. Padding the incoming message to a length multiple of 512 bits. The supplemented message is divided into blocks of 512 bits each.

Step 2. Binary values are generated to fill registers *c* and *r* with a total length of 1600 bits, with *c* being 1024 bits long and filled with logical ones, and the remaining 576 bits (making up *r*) filled with logical zeros. It is worth noting that the architecture provides for the possibility of using a hash function as a message authentication code. Then the registers are filled with a private key or an initialization vector.

Step 3. The resulting register of 1600 bits is considered as a one-dimensional cellular automaton and is subjected to 200 cycles of transformations according to the elementary rule "86" (*R*86).

The result of such transformations will be the starting state of the register, ready for mixing the incoming message.

Step 4. The "absorption" mode begins. The current 512-bit block of the input message is added modulo two to the bits of the register: $RC = RC$ XOR M_i[512].

Step 5. Thirty rounds of shuffling are introduced:

– $RC = R86(RC)$: the *RC* state is processed according to the "86" rule;

– $RC' = R150(RC)$: the *RC* state is processed according to the "150" rule, and the result is placed in the *RC* buffer array';

– $RC >>> 31$: 31-bit cyclic shift of the original state to the right;

– $RC' = RC$ XOR RC': the elements of the *RC'* array are added modulo two to the original *RC* state, the result is in the buffer array;

– $RC = R150(RC')$: the buffer array is processed according to the "150" rule and copied to the original RC state.

Steps 4 and 5 are repeated until the input message blocks are completely exhausted.

Step 6. The "squeezing" stage begins. Each time after the end of the thirtieth round of processing, another 64 bits of the hash image are sent to the output, which are concatenated with the previous ones, so that the resulting length should be a multiple of 64 bits. Since the processing of the contents of the registers continues at the "squeeze" stage, the authors of the architecture claim that it is collision-free.

To investigate the avalanche effect, the developed hashing function was implemented in code in the Java programming language. 64-, 128-, 256-, 512-, and 1024-bit hash images were investigated. A known expression was used as an input message "*The quick brown fox jumps over the lazy dog*" without a period and with a period at the end. The result for a 256-bit hash image is:

– no period:

«0xbed39b7aae3694fe2d4a57df88327589b6670c68dc9c13d391166865234cfcf»;

– with a dot:

«0x2a1a2666518c676a28e587450dccf620198805800901 35f590d7640ff337382».

Obviously, the obtained hash images are completely different. Similar results were obtained for the rest of the studied image sizes.

The obtained results of the avalanche effect study, as well as statistical testing, demonstrated that the developed hashing function can be successfully used in cryptographic applications.

5.4 A BLOCK CIPHER BASED ON THREE-DIMENSIONAL CAS

This cipher uses three-dimensional CAs to process both the input block and the key.

The main parameters of the developed cipher are as follows:

– block size – 64 bytes (512 bits);

– key size – 64 bytes (512 bits);

– number of rounds – up to 15;

– a whole block is processed in one round.

The specified length of the block and key, in our opinion, is optimal for modern cryptographic needs. The optimal number of rounds will be established after conducting statistical tests.

The construction of a three-dimensional 64-byte CA was reduced to the selection of non-equivalent elementary statistically independent rules of interaction, which will be applied to the nearest neighbors of the cell in three directions: X (rule “105”), Y (rule “22”) and Z (rule “150”). At the same time, the state of one cell is expressed by a byte, the cell interacts with 6 neighbors (two neighbors each along the X, Y, Z axes). Each pair of neighbors together with the cell form a separate elementary CA. The new state of the cell is formed after applying all three rules to it in the following order: X – Y – Z. Bitwise operations on the byte are performed by the built-in tools of the programming language. As a cipher architecture, let’s use an SP network similar to the AES cipher. The block diagram of the encryption and decryption processes of one block is shown in **Fig. 5.12**.

Let’s describe the encryption and decryption procedures used.

TransformDataCube. The operation is responsible for the reversible transformation of the data block. In this operation, let’s use a simple XOR operation, since it is self-reversible, with 8 nearest neighbors lying on the four diagonals of the cube.

Accordingly, the reverse operation when decrypting the block will be the same.

RotateCubeMatrices. This operation is analogous to the shiftRows function of the Rijndael algorithm for a three-dimensional block of data. In this case, the four layers of the 4×4×4 cube are rotated as follows: the top (zero) layer is not rotated, the next (first) layer is rotated counterclockwise by 90 degrees, the second layer is rotated by 180 degrees, and the bottom (third) layer is rotated by 270 degrees. It resembles the turns that are implemented in a 4×4×4 Rubik’s cube. When decoding a block of data, the inv operation is performed *RotateCubeMatrices*, where the rotations are clockwise.

SubBytes. This operation performs a permutation using S-boxes of the Rijndael cipher, which have proven to be highly non-linear and crypto-resistant. The direct substitution table is used by the SubBytes operation, the inverse one is used by invSubBytes during decryption. The forward and reverse substitution tables can be found in FIPS-197, which describes the AES standard.

XorRoundKey. A round key scrambling operation, a simple XOR of the bytes of the data block and the key, the result of which is written to the data block. Accordingly, the same operation is used for decryption.

Key deployment operation

This operation does not require reversibility, so it is simply necessary to generate the volume of key material required for a certain number of rounds. Since the key is 64 bytes long (512 bits), for the number of NR rounds it is necessary to generate 64×NR bytes of key material (512×NR in bit terms). So, for a 15-round cipher, it is necessary 960 bytes (7680 bits) of key material. All keys are generated once before the start of encryption / decryption and are placed in a list from which the key needed in the next round is selected.

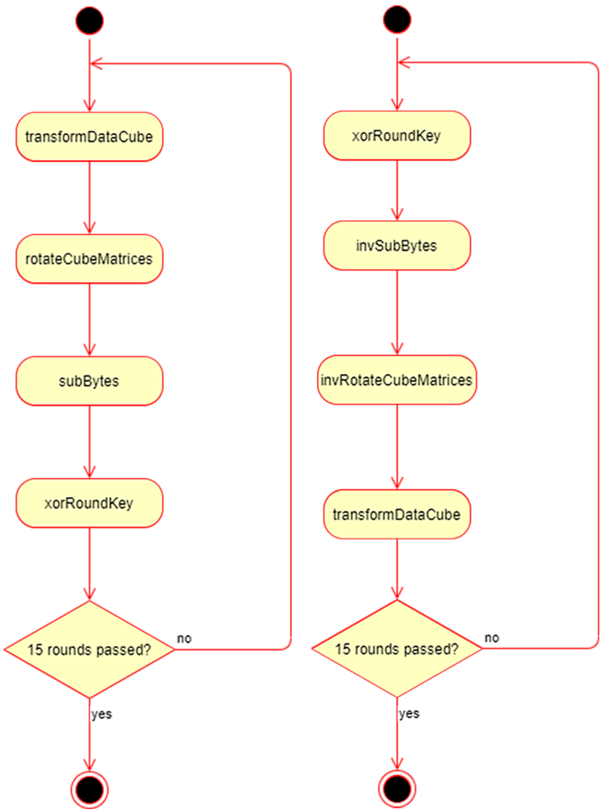


Fig. 5.12 Flowcharts of encryption (left) and decryption of one input block

To generate round keys from the encryption key, the same CA is used as for the input unit with the same inter-cell communication functions. At the beginning, the encryption key bytes are placed

in the CA. Each application of the inter-cell interaction rules sequentially gives another round key that is unloaded into the list. In this way, the requirement of round key synchronization on both sides of the data transmission system is satisfied.

Crypto-resistance of the algorithm testing

Testing the statistical characteristics of ciphers has already become a de facto standard in the development of various cryptographic algorithms. For this purpose, the NIST STS statistical package [129] is used, which consists of 188 different statistical tests grouped into 16 groups.

The testing was carried out in the following way. The system under study encrypts an input file of size 12.5 MB (which is 100 million bits). The file is encrypted on one key with different number of rounds (from 1 to 15). The encrypted sequence is submitted to the NIST STS input and, after receiving the test results, it is possible to draw conclusions about the optimal number of rounds and, in general, about the cryptographic stability of the developed algorithm according to NIST criteria.

It is advisable to present the results obtained in this way in tabular form (**Table 5.2**).

Table 5.2 Results of round statistical testing of the block cipher

Rounds, k't		Proportion, %							
		100	**99**	**98**	**97**	**96**	**95**	**<95**	**Average value**
1	Number of passed tests	71	67	39	7	3	1	-	0.99027
2		73	58	39	8	10	-	-	0.98936
3		60	64	44	18	-	1	1	0.98846
4		65	68	31	16	3	4	1	0.98840
5		76	60	39	12	1	-	-	0.99053
6		72	56	40	14	5	-	1	0.98909
7		67	59	39	12	10	1	-	0.98840
8		71	72	40	5	-	-	-	0.99112
9		75	60	36	11	3	3	-	0.98979
10		77	66	29	13	2	1	-	0.99064
11		75	55	35	11	11	1	-	0.98899
12		69	60	40	13	4	2	-	0.98909
13		83	49	39	11	6	-	-	0.99021
14		86	50	38	13	1	-	-	0.99101
15		61	64	39	16	3	5	-	0.98793

It is convenient to present tabular results in graphic form (**Fig. 5.13**).

As can be seen from the Table and Figure, the majority of statistical tests were passed at a level greater than 0.95. Nevertheless, it can be seen that the number of rounds has a different effect

on the statistical characteristics of the cipher, although the difference is small. As the number of rounds increases, the average value of passing the tests is maximum in the 8th and 14th rounds. At 15 rounds, the average value is the smallest, although it differs from the maximum (at 8 rounds) by only 0.3 %. Moreover, at 15 rounds, the system fails 5 tests. According to the presented results, it is possible to conclude that the 8-round variant of the cipher is optimal both from the point of view of crypto-resistance and speed, in which the maximum average value when passing all tests is observed at a level not lower than 0.97. Good results were also demonstrated with 14 rounds.

In general, as a result of the research, it can be concluded that the developed block cipher based on the three-dimensional CA demonstrated a level of cryptoresistance sufficient for its use.

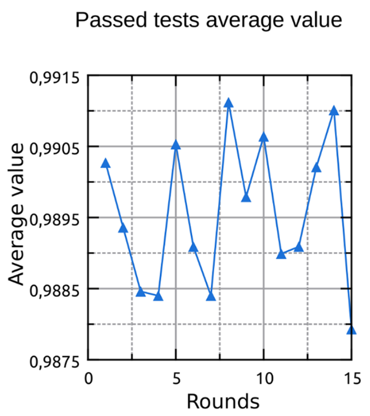


Fig. 5.13 Graph of the average values of passing the NIST STS tests

A stream cipher based on cellular automata

An alternative to block ciphers for encrypting communication channels' connection can be stream ciphers based on crypto-resistant generators of pseudo-random / random sequences. Sequences generated by such a hardware, software, or hardware-software generator are added bit by bit to the open message and form a cipher text whose properties are fully ensured by the cryptographic stability of the generated sequence.

Often, to build a high-quality generator, random sequences are used, which can be obtained from the hardware part of a personal computer. In such cases, it is best not to use hardware generators that are built into the microprocessor, but to generate a hardware sequence based on user actions: to record the time of pressing keys on the keyboard, the time interval between successive key presses, or the time of changing the direction of mouse movement, etc.

We have developed the appropriate software that allows the user to generate such a sequence, and works in three modes with the possibility of their combination:

1) measurement of keyboard keystroke time (ms);

2) measuring the time interval between keyboard keystrokes (ms);

3) time to change the direction of the computer mouse cursor (ms).

The hardware "entropy" generated in this way was sequentially hashed by a cryptographic hashing function based on cellular automata. As a result, let's get a hardware-software key gamma generator that can be used to create a stream cipher.

To build the hashing function, let's use the "cryptographic sponge" architecture proposed by the authors of the SHA-3 hash function [130, 131], which is presented in **Fig. 5.11**.

The mode of operation of such a structure consists of two stages: "absorbing", when the input information is added in blocks to the shuffling function *f*, and "squeezing", when the result of hashing by concatenated blocks is fed to the output of the hashing function. Thus, the required length of the resulting hash image is achieved. Let's use a self-developed shuffle function based on cellular automata.

The working array $r+c$, which is called "state" and has a size of 1600 bits, is filled with the input vector M generated by the hardware and padded with the required number of zeros $M||0^k$, where $k=[r+c]-[M]$. Thus, there are 25 elements of type ULONG, that is, 64-bit unsigned integers.

The array (matrix $A[5\times5]$) is divided into 5 intermediate blocks, the values of which are calculated using the XOR operation as follows:

$$B[0]=A[0]\oplus A[5]\oplus A[10]\oplus A[15]\oplus A[20],$$

$$B[1]=A[21]\oplus A[22]\oplus A[23]\oplus A[24]\oplus A[19],$$

$$B[2]=A[14]\oplus A[9]\oplus A[4]\oplus A[3]\oplus A[2],$$

$$B[3]=A[1]\oplus A[6]\oplus A[11]\oplus A[16]\oplus A[17],$$

$$B[4]=A[18]\oplus A[13]\oplus A[8]\oplus A[7]\oplus A[12].$$

Elements of the array $B[i]$ are used to build additional elements $C[i]$ in the following way: $c[i]=\left(B[j]\ll i\right)\oplus B[j+1]$, where $i=0\text{-}14$, $j=i \bmod 5$.

As a result of such calculations, it is possible to fill the state matrix with new elements. Filling is performed by columns: the first – A[0]–$A[4]$; the second – $B[0]$–$B[4]$; the remaining columns are successively filled with elements of C[i].

At the second stage of mixing the state matrix, the elementary rules of cellular automata are used. These are rules "30" and "146". They are chosen because they exhibit chaotic and unpredictable behavior, which is crucial for the hashing function. First, the "30" rule is used, and the next pass – the "146" rule. Passes are performed spirally clockwise: $A[i] = ((A[i])_{R30})_{R146}$. Here, the indices R30 and R146 indicate the application of elementary rules "30" and "146" of cellular automata to the elements of the array.

To improve the avalanche effect, the final permutation of the state elements is performed with a cyclic shift of bits by a certain number of positions: $A[i] = Sh_K(A[i+3])$, $K \in \{7,11,13,23,29\}$, and Sh_K is a cyclic shift of bits to the left by K positions, $i = 0$–24.

After the "absorbing" stage, the "squeezing" stage is started, the result of which is a generated random stream that can be used for stream encryption.

Results of statistical testing

To evaluate the quality of the developed hashing function (as well as the entire stream cipher), let's use the NIST STS statistical package. Testing was carried out similarly to the previous one. The results of statistical testing of the hash function developed by us were compared with the standard SHA-3 hashing function (**Table 5.3**).

Table 5.3 Results of statistical testing of hash functions

Test passing rate, %	SHA-3 (Keccak) based on cellular automata	Classic SHA-3 (Keccak)
100	79 (42.02 %)	58 (31.01 %)
99	64 (34.04 %)	68 (36.36 %)
98	27 (14.3 %)	45 (24.06 %)
97	15 (7.97 %)	13 (6.95 %)
96	3 (1.65 %)	1 (0.53 %)
95	0 (0 %)	1 (0.53 %)
<95	0 (0 %)	1 (0.53 %)

As can be seen from the table, the statistical results of the hashing function developed by us are at least not worse than those of the classic SHA-3. To evaluate the avalanche effect, three variants of the well-known pangram phrase "The quick brown fox jumps over the lazy dog", which contains all the letters of the English alphabet, were chosen:

1) *The quick brown fox jumps over the lazy dog;*
2) *The quick brown fox jumps over the lazy dog. (with a period at the end);*
3) *the quick brown fox jumps over the lazy dog (from a small letter).*

Below are the resulting hash values of the algorithm we developed based on cellular automata:

1) *42063cf95300891402c6cb0913c788ab50bf8c2cd2f4dc1781fdebafcfb-7679928c60d41609fd80165b7e63eebdc78ab7656a89af1e5e218a87cc3cf475be4d1;*

2) *56b949ac21cf22a78c1d92bf01ef20937f8db80083a84cff1d0879b9b54d6fd-9b787a0ab4539b8bfb0e57c002749bd5a8fdf37c2777b3cceece256a53df66a51;*

3) *ce8f6d5fc67cd52933ccba3836f9470b32a896a9545682a9850447df691ec-39c59af4504547020e885d5ffef192eb1baa7af650ca7dd30413929c12f233f25b3.*

As it is possible to see, adding a single character or changing an uppercase letter to lowercase completely changes the hash value, which is a sign of a good avalanche effect.

Thus, the proposed methods can be successfully applied to generate a binary random sequence based on hardware and software solutions. **Fig. 5.14** shows a diagram of the results of statistical testing of the developed binary random flow generator based on cellular automata. It is possible to see that the proposed method gives good statistical results.

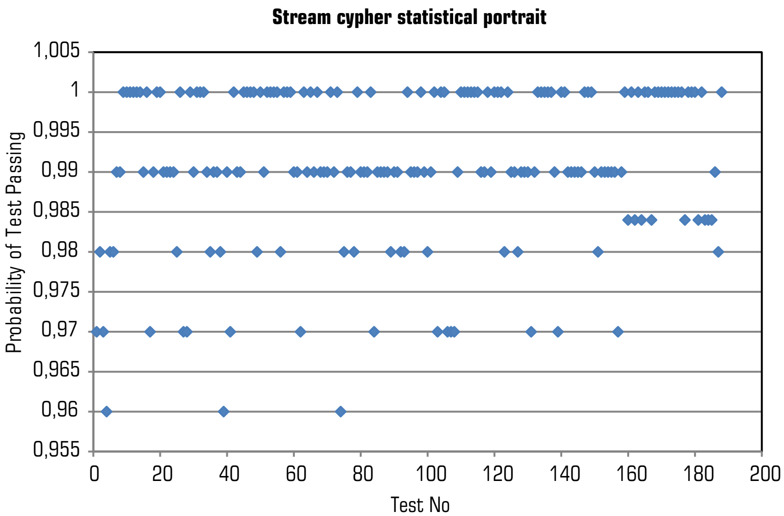


Fig. 5.14 Results of statistical testing of the developed binary random sequence generator based on cellular automata. The abscissa shows the number of the test, the ordinate shows the probability of passing it

As can be seen from the figure, the stream cipher developed by us based on CA passes all NIST STS tests, which indicates the adequacy of the applied transformations, including those based on cellular automata. Thus, it is possible to recommend this cipher to protect the confidentiality of data when transmitted over telecommunications systems.

CONCLUSIONS

1. Hybrid (cyber-physical / socio-cyber-physical) information systems are boosted in the conditions of rapid development of computing resources and technologies, integration of various components of high technologies. This requires a new approach to providing not only security services, but also the multi-loop protection systems. The proposed concept of determining the level of security is based on the concept of a critical business process and takes into account the points of execution of this business process, as well as the hybridity and synergy of modern threats.

2. Sets of rules for determining the achievability of a given level of security based on estimates of the integrity, availability and confidentiality of information arrays, as well as computer technology relative to various points of the organization's business processes, have been formed. This approach provides an objective link between business goals, infrastructure objects / elements of hybrid information systems, security services, taking into account the requirements of regulators, the goals and functionality of critical (continuous) business processes of a company / organization / enterprise. A system for assessing the level of system security has been developed, implemented in the declarative programming language Prolog, which, in dialogue with the user, generates a response on the achievability of a given level of system security, depending on the assessments of the state of individual system components reported to it.

3. The proposed mathematical model for constructing asymmetric cryptosystems based on McEliece and Niederreiter CCC makes it possible to provide the required level of confidentiality, integrity and authenticity services and to practically implement the proposed method. This approach provides the required level of protection of security services, and the use of various noise-immune codes allows, taking into account the level of information secrecy, to ensure its reduction in energy consumption and increase the efficiency of information transmission.

4. The models of probable threats and protection of information in public networks are proposed. It is noted that the most general model of the formal description of the protection system is the model of the security system with full overlap, which defines a complete list of objects of protection and information threats, means of ensuring security from the point of view of their effectiveness and contribution to ensuring the security of the entire telecommunications system. The proposed model satisfies the general scheme of interactions, which includes the organizational management system, external and internal threats, as well as the environment of interaction between them. It follows from the comparison of the model with a complete overlap and the general scheme of interaction that the model does not take into account the possibility of simultaneous implementation of different types of threats, the power of influence, the possibilities of their interaction and, accordingly, preventive protective measures. It is concluded that the development of the model of the formal description of the protection system is connected, first of all, with the consideration of the detailed description of the object of protection with the introduction of a more

general concept of threats, as well as the construction of an apparatus for modeling various types of threats and their interactions.

It is shown that the combination of four models (passive and active channel of information leakage, unintentional and unauthorized access to information for the purpose of its removal) in various variants provides wide opportunities for modeling various known types of threats and their implementation. It is noted that it is necessary to use such approaches to ensuring information protection that allow detection and prevention of threats of unknown types and dynamic correction of protection behavior, adapting it to specific application conditions.

It is proposed to use a special channel in its structure to take into account unknown types of threats that are subject to identification and adaptation contour. The conditions under which the information transmission system will turn into a feedback system, where the transmission channel affects the response of the system, are listed. The information protection model provides an opportunity for constant refinement of threat classes and response measures and continuous training of the adaptive component of the CSI. Thus, the CSI, built on the basis of this model, detects and prevents threats of unknown types.

A model of unauthorized access to information for the purpose of its removal with an unknown type of threat and an unspecified level of protection is also proposed. Its structure includes: a special module of internal diagnostics, which diagnoses the entire protection system, makes a decision to adjust the algorithm of the SHI behavior, which allows to achieve fault tolerance of the information security tools. The use of a special module that diagnoses the communication channel with subsequent changes in the level of security allows to achieve the adaptability of the information security tools.

5. An experimental study of the functioning of modern means of exploiting vulnerabilities was conducted, in particular, the process of checking and confirming the possibility of implementing vulnerabilities was investigated. Based on the observations, the general characteristics of the vulnerability validation process were identified, which take into account the complex and changing nature of the environment, as well as the risk of a critical error in the functioning of the target system during the exploitation of vulnerabilities. Variable – the number of successfully validated vulnerabilities on the target system (success validation), – number of unvalidated vulnerabilities (failed validation), – the number of cases of vulnerability validation that led to a critical error in the target system and subsequent loss of communication with it (crash validation). A regression analysis of the obtained results of experimental studies was carried out and a mathematical model was developed for the analysis of quantitative characteristics of the vulnerability validation process based on Bernstein polynomials, which allow describing the dynamics of this process.

Analytical dependencies were obtained for the number of successfully validated and unvalidated vulnerabilities, as well as for the number of cases of vulnerability validation that led to critical errors during the rational cycle of validation of detected vulnerabilities during active analysis of corporate network security. The proposed dependencies make it possible to build probability distribution laws for the above-mentioned characteristics of the vulnerability validation process.

On the basis of the conducted practical analysis of the vulnerability validation process and the obtained analytical dependencies of the basic characteristics of the validation process, the quality indicators of the mechanism of validation of corporate network vulnerabilities were selected and characterized.

A technique for analyzing the quality of work of the mechanism for validating detected vulnerabilities of the corporate network has been developed, which is based on integral equations that take into account the quantitative characteristics of the investigated mechanism of validation of vulnerabilities at a certain point in time. This technique allows to build the distribution laws of the quality indicators of the vulnerability validation process and to quantitatively evaluate the quality of the mechanism of validation of identified vulnerabilities, which in turn allows to monitor and control the progress of validation of identified vulnerabilities during active security analysis in real time.

A method of building a fuzzy knowledge base for decision-making in the validation of vulnerabilities of software and hardware platforms during an active analysis of the security of the target corporate network is proposed, based on the use of fuzzy logic, which makes it possible to obtain reliable information about the quality of the vulnerability validation mechanism in an indirect way. The built knowledge base allows for the formation of decisive decision-making rules regarding the implementation of one or another attacking action, which in turn allows for the development of expert systems for automating the decision-making process during the validation of identified vulnerabilities of target information systems and networks.

The method of automatic active analysis of security has received further development, which, based on the synthesis of the proposed model, methodology and method, allows, unlike the existing ones, to abstract from the conditions of dynamic changes in the environment, that is, the constant development of information technologies, which leads to an increase in the number of vulnerabilities and corresponding vectors attacks, as well as the growth of ready-to-use vulnerability exploits and their availability, and consider only the quality parameters of the vulnerability validation process itself.

6. The statistical characteristics of generators of pseudorandom sequences, block and stream ciphers based on cellular automata were developed and studied. It is shown that they can be successfully used for cryptographic applications as efficient means of scrambling incoming messages. The developed generators based on a combination of intercellular interaction rules and an original self-developed rule demonstrated good statistical properties, which was confirmed by NIST STS tests. A 3-dimensional cellular automaton based on the rules "22", "105" and "150" was used to develop the block cipher. Substitution tables from the well-known standard AES cipher are used. Developed own round function and key unwrapping operation.

A hardware and software generator of a random binary sequence based on cellular automata has also been developed.

The keyboard and mouse of a personal computer were used as a hardware platform for generating input entropy. For further processing of the received random sequence, a proprietary hashing function was developed in the "cryptographic sponge" architecture. A proprietary mixing function has been developed, where elementary rules of CA "30" and "146" are applied. Statistical testing

of the resulting cryptographic applications was performed using the NIST STS statistical package, which demonstrated good statistical performance of both ciphers. Studies have also shown a good avalanche effect of the designed hashing function based on CA. Summing up, it is possible to confidently state that the use of cellular automata, both one-dimensional and multidimensional, enables engineers to construct simple and effective cryptographic structures, resulting in high-quality means of protecting the confidentiality and integrity of information.

REFERENCES

1. Milov, O., Khvostenko, V., Natalia, V., Korol, O., Zviertseva, N. (2022). Situational Control of Cyber Security in Socio-Cyber-Physical Systems. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). doi: https://doi.org/10.1109/hora55278.2022.9800049
2. Yevseiev, S., Ponomarenko, Rayevnyeva, O. (2017). Assessment of functional efficiency of a corporate scientificeducational network based on the comprehensive indicators of quality of service. Eastern-European Journal of Enterprise Technologies, 6 (2 (90)), 4–15. doi: https://doi.org/10.15587/1729-4061.2017.118329
3. Androshchuk, A., Yevseiev, S., Melenchuk, V., Lemeshko, O., Lemeshko, V. (2020). Improvement of project risk assessment methods of implementation of automated information components of non-commercial organizational and technical systems. EUREKA: Physics and Engineering, 1, 48–55. doi: https://doi.org/10.21303/2461-4262.2020.001131
4. Pospelov, D. A. (2021). Situatuinal control: Theory and Practice. URSS, 288.
5. Li, T., Liu, J., Sun, H., Chen, X., Yin, L., Mao, X., Sun, J. (2021). Runtime Verification of Spatio-Temporal Specification Language. Mobile Networks and Applications, 26 (6), 2392–2406. doi: https://doi.org/10.1007/s11036-021-01779-5
6. Lange, R. T. (2019). Action Grammars: A Grammar Induction-Based Method for Learning Temporally-Extended Actions. Imperial College London.
7. Kempson, R., Cann, R., Gregoromichelaki, E., Chatzikyriakidis, S. (2017). Action-Based Grammar. Theoretical Linguistics, 43 (1-2), 141–167. doi: https://doi.org/10.1515/tl-2017-0012
8. Milov, O., Yevseiev, S., Vlasov, A., Herasimov, S., Dmitriiev, O., Kasianenko, M. et al. (2019). Development of scenario modeling of conflict tools in a security system based on formal grammars. Eastern-European Journal of Enterprise Technologies, 6 (9 (102)), 53–64. doi: https://doi.org/10.15587/1729-4061.2019.184274
9. Mehrabian, M., Khayatian, M., Shrivastava, A., Eidson, J. C., Derler, P., Andrade, H. A. et al. (2017). Timestamp Temporal Logic (TTL) for Testing the Timing of Cyber-Physical Systems. ACM Transactions on Embedded Computing Systems, 16 (5s), 1–20. doi: https://doi.org/10.1145/3126510
10. Yevseiev, S., Milov, O., Opirskyy, I., Dunaievska, O., Huk, O., Pogorelov, V. et al. (2022). Development of a concept for cybersecurity metrics classification. Eastern-European Journal of Enterprise Technologies, 4 (4 (118)), 6–18. doi: https://doi.org/10.15587/1729-4061.2022.263416
11. Abel, S., Xiao, L., Wang, H. (2018). Causal Modeling for Cybersecurity. 2018 International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec), 209–212. doi: https://doi.org/10.1109/socialsec.2018.8760379

13. Kondakci, S. (2010). A causal model for information security risk assessment. 2010 Sixth International Conference on Information Assurance and Security, 143–148. doi: https://doi.org/10.1109/isias.2010.5604039
14. Friedenberg, M., Halpern, J. Y. (2018). Combining the causal judgments of experts with possibly different focus areas. Available at: http://www.cs.cornell.edu/home/halpern/papers/focus.pdf
15. Halpern, J. Y. (2015). A modification of the Halpern-Pearl definition of causality. International Joint Conference on Artificial Intelligence, 3022–3033. Available at: https://www.ijcai.org/Proceedings/15/Papers/427.pdf
16. Fenz, S., Ekelhart, A. (2011). Verification, Validation, and Evaluation in Information Security Risk Management. IEEE Security & Privacy Magazine, 9 (2), 58–65. doi: https://doi.org/10.1109/msp.2010.117
17. IEC 31010:2019 (2019). Risk management – Risk assessment techniques. International Organization of Standardization (ISO), 264.
18. Shaikh, F. A., Siponen, M. (2023). Information security risk assessments following cybersecurity breaches: The mediating role of top management attention to cybersecurity. Computers & Security, 124, 102974. doi: https://doi.org/10.1016/j.cose.2022.102974
19. Haag, S., Siponen, M., Liu, F. (2021). Protection Motivation Theory in Information Systems Security Research: A Review of the Past and a Road Map for the Future. ACM SIGMIS Database: The DATABASE for Advances in Information Systems, 52 (2), 25–67. doi: https://doi.org/10.1145/3462766.3462770
20. Li, Y., Xin, T., Siponen, M. (2022). Citizens' Cybersecurity Behavior: Some Major Challenges. IEEE Security & Privacy, 20 (1), 54–61. doi: https://doi.org/10.1109/msec.2021.3117371
21. Chen, S., Xiao, H., He, W., Mou, J., Siponen, M., Qiu, H., Xu, F. (2021). Determinants of Individual Knowledge Innovation Behavior. Journal of Organizational and End User Computing, 33 (6), 1–24. doi: https://doi.org/10.4018/joeuc.20211101.oa27
22. Yevseiev, S., Kuznietsov, O., Herasimov, S., Horielyshev, S., Karlov, A., Kovalov, I. et al. (2021). Development of an optimization method for measuring the Doppler frequency of a packet taking into account the fluctuations of the initial phases of its radio pulses. Eastern-European Journal of Enterprise Technologies, 2 (9 (110)), 6–15. doi: https://doi.org/10.15587/1729-4061.2021.229221
23. Yevseiev, S., Hryshchuk, R., Molodetska, K., Nazarkevych, M., Hrytsyk, V., Milov, O. et al.; Yevseiev, S., Hryshchuk, R., Molodetska, K., Nazarkevych, M. (Eds.) (2022). Modeling of security systems for critical infrastructure facilities. Kharkiv: PC TECHNOLOGY CENTER, 196. doi: http://doi.org/10.15587/978-617-7319-57-2
24. ISO/IEC 27003:2017 (2017). Information technology – Security techniques – Information security management systems – Guidance. Available at: https://www.iso.org/ru/standard/63417.html
25. ISO/IEC 27006:2015 (2015). Information technology – Security techniques – Requirements for bodies providing audit and certification of information security management systems. Available at: https://www.iso.org/standard/62313.html

26. ISO / IEC 27035-1:2023 (2023). Information technology – Information security incident management – Part 1: Principles and process. Available at: https://www.iso.org/ru/standard/78973.html
27. ISO/IEC 27035-2:2023 (2023). Information technology – Information security incident management – Part 2: Guidelines to plan and prepare for incident response. Available at: https://www.iso.org/ru/standard/78974.html
28. ISO/IEC 27035-3:2020 (2020). Information technology – Information security incident management – Part 3: Guidelines for ICT incident response operations. Available at: https://www.iso.org/ru/standard/74033.html
29. Lund, M. S., Solhaug, B., Stølen, K. (2010). Model-Driven Risk Analysis: The CORAS Approach. Springer Publishing Company, Incorporated, 400.
30. Matulevičius, R. (2017). Domain Model for Information Systems Security Risk Management. Fundamentals of Secure System Modelling. Springer International Publishing AG, 17–30. doi: https://doi.org/10.1007/978-3-319-61717-6_2
31. Innerhofer-Oberperfler, F., Mitterer, M., Hafner, M., Breu, R. (2010). Security Analysis of Service Oriented Systems: A Methodical Approach and Case Study. Web Services Security Development and Architecture. IGI Global, 33–56. doi: https://doi.org/10.4018/978-1-60566-950-2.ch002
32. Innerhofer-Oberperfler, F., Breu, R.; Moore, T., Pym, D., Ioannidis, C. (Eds.) (2010). Potential Rating Indicators for Cyberinsurance: An Exploratory Qualitative Study. Economics of Information Security and Privacy. Boston: Springer. doi: https://doi.org/10.1007/978-1-4419-6967-5_13
33. Alkubaisy, D., Piras, L., Al-Obeidallah, M. G., Cox, K., Mouratidis, H.; Ali, R., Kaindl, H., Maciaszek, L. A. (Eds.) (2022). A Framework for Privacy and Security Requirements Analysis and Conflict Resolution for Supporting GDPR Compliance Through Privacy-by-Design. Evaluation of Novel Approaches to Software Engineering. Cham: Springer, 67–87. doi: https://doi.org/10.1007/978-3-030-96648-5_4
34. Pullonen, P., Tom, J., Matulevičius, R., Toots, A. (2019). Privacy-enhanced BPMN: enabling data privacy analysis in business processes models. Software and Systems Modeling, 18 (6), 3235–3264. doi: https://doi.org/10.1007/s10270-019-00718-z
35. Malina, L., Dzurenda, P., Ricci, S., Hajny, J., Srivastava, G., Matulevicius, R. et al. (2021). Post-Quantum Era Privacy Protection for Intelligent Infrastructures. IEEE Access, 9, 36038–36077. doi: https://doi.org/10.1109/access.2021.3062201
36. Koeze, R. (2017). Designing a Rule-Based Cyber Risk Assessment Tool for Small to Medium Enterprises. Delft University of Technology. The TREsPASS Project. The TREsPASS Project.
37. Pohasii, S., Yevseiev, S., Zhuchenko, O., Milov, O., Lysechko, V., Kovalenko, O. et al. (2022). Development of crypto-code constructs based on LDPC codes. Eastern-European Journal of Enterprise Technologies, 2 (9 (116)), 44–59. doi: https://doi.org/10.15587/1729-4061.2022.254545

38. Shmatko, O., Balakireva, S., Vlasov, A., Zagorodna, N., Korol, O., Milov, O. et al. (2020). Development of methodological foundations for designing a classifier of threats to cyberphysical systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (105)), 6–19. doi: https://doi.org/10.15587/1729-4061.2020.205702
39. Merz, H., Hansemann, T., Hübner, C. (2009). Building Automation: Communication systems with EIB / KNX, LON und BACnet. Berlin Heidelberg: Springer-Verlag, 293.
40. KNX Technical Manual 2CKA001473B8668. KNX Technical Manual. Busch-Presence detector KNX / Busch-Watchdog Sky KNX (2017). Busch-Jaeger Elektro GmbH, 198.
41. Technical documentation on KNX devices (2006). ABB.
42. KNX Handbook Version 1.1 Revision 1 (2004). Konnex Association.
43. ABB i-bus KNX KNX Security Panel GM / A 8.1 Product Manual (2016). Busch-Watchdog Sky KNX. Busch-Jaeger Elektro GmbH, 648.
44. Schilder, J., Reibel, T. (2016). ABB GPG Building Automation Webinar ABB i-bus® KNX Basics and Products. Busch-Watchdog Sky KNX. Busch-Jaeger Elektro GmbH, 86.
45. Manual for KNX Planning (2017). Siemens Switzerland Ltd, 100.
46. Security Technology KNX-Intrusion Alarm System L240 Installation, Commissioning, Operation / Busch-Watchdog Sky KNX (2010). Busch-Jaeger Elektro GmbH, 116.
47. Bartock, M., Cichonski, J., Souppaya, M., Smith, M., Witte, G., Scarfone, K. (2016). Guide for cybersecurity event recovery. doi: https://doi.org/10.6028/nist.sp.800-184
48. Security requirements for cryptographic modules (1994). Information Technology Laboratory. Available at: https://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
49. Cichonski, J., Franklin, J. M., Bartock, M. (2017). Guide to LTE security. doi: https://doi.org/10.6028/nist.sp.800-187
50. Kottapalli, N. (2011). Diameter and LTE Evolved Packet System. Corporate Headquarters, 10. Available at: http://go.radisys.com/rs/radisys/images/paper-lte-diameter-eps.pdf
51. Ventura, H. (2002). Diameter – Next generation's AAA protocol. Institutionen för Systemteknik, 66.
52. Vinay, K. S. B., Harihar, M. N. (2012). Diameter-Based Protocol in the IP Multimedia Subsystem. International Journal of Soft Computing and Engineering, 1 (6), 266–269.
53. Qanbari, S., Mahdizadeh, S., Rahimzadeh, R., Behinaein, N., & Dustdar, S. (2016). Diameter of Things (DoT): A Protocol for Real-Time Telemetry of IoT Applications. Lecture Notes in Computer Science. GECON, 207–222. doi: https://doi.org/10.1007/978-3-319-43177-2_14
54. Tschofenig, H. (2019). Diameter: new generation AAA protocol – design, practice, and applications. Hoboken: John Wiley & Sons, Ltd, 230. doi: https://doi.org/10.1002/9781118875889
55. Ugrozy bezopasnosti iadra paketnoi seti 4G (2017). Available at: https://www.ptsecurity.com/ru-ru/research/analytics/epc-2017/
56. Uiazvimosti protokola Diameter v setiakh 4G (2018). Available at: https://www.ptsecurity.com/ru-ru/research/analytics/diameter-2018/

57. Yevseiev, S., Tsyhanenko, O., Ivanchenko, S., Aleksiyev, V., Verheles, D., Volkov, S. et al. (2018). Practical implementation of the Niederreiter modified cryptocode system on truncated elliptic codes. Eastern-European Journal of Enterprise Technologies, 6 (4 (96)), 24–31. doi: https://doi.org/10.15587/1729-4061.2018.150903
58. Yevseiev, S., Korol, O., Kots, H. (2017). Construction of hybrid security systems based on the crypto-code structures and flawed codes. Eastern-European Journal of Enterprise Technologies, 4 (9 (88)), 4–21. doi: https://doi.org/10.15587/1729-4061.2017.108461
59. Yevseiev, S., Kots, H., Liekariev, Y. (2016). Developing of multi-factor authentication method based on niederreiter-mceliece modified crypto-code system. Eastern-European Journal of Enterprise Technologies, 6 (4 (84)), 11–23. doi: https://doi.org/10.15587/1729-4061.2016.86175
60. Yevseiev, S., Rzayev, K., Korol, O., Imanova, Z. (2016). Development of mceliece modified asymmetric crypto-code system on elliptic truncated codes. Eastern-European Journal of Enterprise Technologies, 4 (9 (82)), 18–26. doi: https://doi.org/10.15587/1729-4061.2016.75250
61. Pohasii, S., Yevseiev, S., Zhuchenko, O., Milov, O., Lysechko, V., Kovalenko, O. et al. (2022). Development of crypto-code constructs based on LDPC codes. Eastern-European Journal of Enterprise Technologies, 2 (9 (116)), 44–59. doi: https://doi.org/10.15587/1729-4061.2022.254545
62. Naim, M., Ali-Pacha, H., Ali-Pacha, A., Hadj-Said, N. (2021). Lengthening the period of a Linear Feedback Shift Register. Journal of Engineering Technology and Applied Sciences, 6 (1), 45–68. doi: https://doi.org/10.30931/jetas.778792
63. Khoroshko, V. O., Pavlov, I. M., Bobalo, Y. Ya., Dudykevich, V. B. et al. (2020). Design of complex information protection systems. Lviv: Lviv Polytechnic, 320.
64. Brailovskyi, M. M., Zybin, S. V., Piskun, I. V., Khoroshko, V. O., Khokhlacheva, Yu. E. (2021). Tekhnolohii zakhystu informatsii. Kyiv: Central Committee “Comprint”, 296.
65. Popovsky, V. V., Persikov, A. V. (2006). Zakhyst informatsii v telekomunikatsiinykh systemakh. Kharkiv: SMIT.
66. Dudykevich, V. B., Khoroshko, V. O., Yaremchuk, Yu. E. (2018). Osnovy informatsiinoi bezpeky. Vinnytsia: VNTU, 315.
67. Efficiency Standards for Protecting ACS and Computers from Data Loss Due to EM Emissions and Pickup (1997). Moscow: MOP, 35.
68. Brailovsky, M. M., Zybin, S. V., Kobozeva, A. A., Khoroshko, V. O., Khokhlacheva, Yu. E. (2021). Analysis of cyber security of information systems. Kyiv: FOP Yamchynskyi O. V., 360.
69. Beloshapkin, V. K., Pustovit, S. M., Stepanov, V. D. (2005). Construction of a specialized model of the information system for the purpose of synthesizing a complex SHI. Protection of information, 3, 78–83.
70. Stallings, V. (2001). Cryptography and network protection, principles and practice. House Williams, 672.

71. Chumarin, I. G. (2001). Taina predpryiatyia: chto y kak zashchyshchat. Saint Petersburg: Ed. DNA, 159.
72. Stepanov, V. D., Khoroshko, V. O. (2003). Collection of sciences. Works Information protection of the Scientific Research Institute of GUR, 5, 12–19.
73. Khoroshko, V. O., Kryvoruchko, O. V., Brailovskyi, M. M., Kozyura, V. D., Desyatko, A. M. (2019). Protection of electronic communications systems. Kyiv. national trade and economy Univ, 164.
74. Domarev, V. V. (2004). Bezopasnost informatcionnykh tekhnologii. Sistemnii pokhod. Kyiv: TND "DS", 992.
75. Maksimov, Yu. A., Fillipovskaya, E. A. (1982). Algorithms for solving problems of nonlinear programming. Moscow: MIPhI, 342.
76. Petrov, A. A., Khoroshko, V. A. (2009). Evaluation of the effectiveness of the complex information protection system in public networks. Collection. Science works of KNU named after T. Shevchenko VIKNU, 21, 128–131.
77. Hayvoronsky, M. V., Novikov, O. M. (2009). Bezpeka informatsiino-komunikatsiinykh system. Kyiv: Ed. BHV group, 608.
78. Pichkur, V. V., Sobchuk, V. V. (2021). Mathematical Model and Control Design of a Functionally Stable Technological Process. Journal of Optimization, Differential Equations and Their Applications, 29 (1), 32. doi: https://doi.org/10.15421/142102
79. Sobchuk, V., Pichkur, V., Barabash, O., Laptiev, O., Kovalchuk, I., Zidan, A. (2020). Algorithm of Control of Functionally Stable Manufacturing Processes of Enterprises. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, 206–210, doi: https://doi.org/10.1109/ATIT50783.2020.9349332
80. Sobchuk, V., Olimpiyeva, Y., Musienko, A., Sobchuk, A. (2021). Ensuring the properties of functional stability of manufacturing processes based on the application of neural networks. CEUR Workshop Proceedings, 2845, 106–116.
81. Maksymuk, O. V., Sobchuk, V. V., Salanda, I. P., Sachuk, Yu. V. (2020). A system of indicators and criteria for evaluation of the level of functional stability of information heterogenic networks. Mathematical Modeling and Computing, 7 (2), 285–292. doi: https://doi.org/10.23939/mmc2020.02.285
82. Barabash, O., Tverdenko, H., Sobchuk, V., Musienko, A., Lukova-Chuiko, N. (2020). The Assessment of the Quality of Functional Stability of the Automated Control System with Hierarchic Structure. 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC). Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 158–161. doi: https://doi.org/10.1109/SAIC51296.2020.9239122
83. Sobchuk, V., Kapustyan, O., Pichkur, V., Kapustian, O. (2021). Design of Stable Periodic Regimes for one Class of Hybrid Planar Systems. II International Scientific Symposium "Intelligent Solutions". Kyiv, 89–100.
84. Kapustian, O. A., Kapustyan, O. V., Ryzhov, A., Sobchuk, V. (2022). Approximate Optimal Control for a Parabolic System with Perturbations in the Coefficients on the Half-Axis. Axioms, 11 (4), 175. doi: https://doi.org/10.3390/axioms11040175

85. Sobchuk, V. V. (2019) The method of creating a single information space at a production enterprise with a functionally stable production process. Control, navigation and communication systems, 6 (58), 84–91.
86. Sobchuk, V., Barabash, O., Musienko, A., Laptiev, O., Kozlovskyi, V., Shcheblanin, Y. (2022). Evaluation of Efficiency of Application of Functionally Sustainable Generalized Information System of the Enterprise. 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications. Ankara. doi: https://doi.org/10.1109/hora55278.2022.9799892
87. Sobchuk, V., Zamrii, I., Vlasyk, H., Tsvietkova, Y. (2021). Strategies for management of operation of production centers to provide functionally sustainable technological processes of production. 2021 IEEE 3 nd International Conference on Advanced Trends in Information Theory (ATIT). Kyiv, 61–66.
88. Sobchuk, V., Zamrii, I., Barabash, O., Musienko, A. (2021). Methodology for building a functionally stable intelligent information system of a manufacturing enterprise. Bulletin of Taras Shevchenko National University of Kyiv. Series: Physics and Mathematics, 4, 116–127. doi: https://doi.org/10.17721/1812-5409.2021/4.18
89. Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O., Korol, O., Milevskyi, S. et al.; Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O. (Eds.) (2021). Synergy of building cybersecurity systems. Kharkiv: PC TECHNOLOGY CENTER, 188. doi: http://doi.org/10.15587/978-617-7319-31-2
90. Vlasyk, H., Zamrii, I., Shkapa, V., Kalyniuk, A., Laptieva, T. (2021). The method of solving problems of optimal restoration of telecommunication signals. 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT). Proceedings, 71–75. doi: https://doi.org/10.1109/atit54053.2021.9678649
91. Svynchuk, O., Barabash, O., Nikodem, J., Kochan, R., Laptiev, O. (2021). Image Compression Using Fractal Functions. Fractal and Fractional, 5 (2), 31. doi: https://doi.org/10.3390/fractalfract5020031
92. Salanda, I. P., Barabash, O. V., Musienko, A. P. (2017). System of indicators and criteria for formalization of processes of ensuring local functional stability of extensive information networks. Systems of control, navigation and communication, 1 (41), 122–126.
93. Petrivskyi, V., Shevchenko, V., Yevseiev, S., Milov, O., Laptiev, O., Bychkov, O. et al. (2022). Development of a modification of the method for constructing energy-efficient sensor networks using static and dynamic sensors. Eastern-European Journal of Enterprise Technologies, 1 (9 (115)), 15–23. doi: https://doi.org/10.15587/1729-4061.2022.252988
94. Lenkov, S., Zhyrov, G., Zaitsev, D., Tolok, I., Lenkov, E., Bondarenko, T. et al. (2017). Features of modeling failures of recoverable complex technical objects with a hierarchical constructive structure. Eastern-European Journal of Enterprise Technologies, 4 (4 (88)), 34–42. doi: https://doi.org/10.15587/1729-4061.2017.108395
95. Yevseiev, S., Rzayev, K., Laptiev, O., Hasanov, R., Milov, O., Asgarova, B. et al. (2022). Development of a hardware cryptosystem based on a random number generator with two

types of entropy sources. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 6–16. doi: https://doi.org/10.15587/1729-4061.2022.265774

96. Mashkov, V. A., Barabash, O. V. (1998) Self-checking and Self-diagnosis of Module Systems on the Principle of Walking Diagnostic Kernel. Engineering Simulation. Amsterdam: OPA, 15, 43–51.
97. Mashkov, V. A., Barabash, O. V. (1996) Self-Testing of Multimodule Systems Based on Optimal Check-Connection Structures. Engineering Simulation. Amsterdam: OPA, 13, 479–492.
98. Mashkov, V. A., Barabash, O. V. (1995). Self-checking of modular systems under random performance of elementary checks. Engineering Simulation. Amsterdam: OPA, 12 (3), 433–445.
99. Barabash, O. V., Dakhno, N. B., Shevchenko, H. V., Majsak, T. V. (2017). Dynamic Models of Decision Support Systems for Controlling UAV by Two-Step Variational-Gradient Method. Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 108–111. doi: https://doi.org/10.1109/apuavd.2017.8308787
100. Mukhin, V., Zavgorodnii, V., Barabash, O., Mykolaichuk, R., Kornaga, Y., Zavgorodnya, A., Statkevych, V. (2020). Method of Restoring Parameters of Information Objects in a Unified Information Space Based on Computer Networks. International Journal of Computer Network and Information Security, 12 (2), 11–21. doi: https://doi.org/10.5815/jcnis.2020.02.02
101. Mukhin, V., Loutskii, H., Barabash, O., Kornaga, Y., Steshyn, V. (2015). Models for Analysis and Prognostication of the Indicators of the Distributed Computer Systems' Characteristics. International Review on Computers and Software (IRECOS), 10 (12), 1216–1224. doi: https://doi.org/10.15866/irecos.v10i12.8023
102. Laptiev, O., Shuklin, G., Savchenko, V., Barabash, O., Musienko, A., Haidur, H. (2019). The Method of Hidden Transmitters Detection based on the Differential Transformation Model. International Journal of Advanced Trends in Computer Science and Engineering, 8 (6), 2840–2846. doi: https://doi.org/10.30534/ijatcse/2019/26862019
103. Sobchuk, V., Barabash, O., Musienko, A., Svynchuk, O. (2021). Adaptive accumulation and diagnostic information systems of enterprises in energy and industry sectors. E3S Web of Conferences, 250, 08002. doi: https://doi.org/10.1051/e3sconf/202125008002
104. Kyrychok, R., Shuklin, G. (2020). The method of building a knowledge base for dicision-making when validating corporate networks vulnerabilities. Scientific Discussion, 1 (47), 7–11.
105. Kyrychok, R. V., Shuklin, G. V. (2020). Methodology for analysing the quality of the vulnerability validation mechanism in the corporate networks. Telecommunication and Information Technologies, 69 (2), 29–40. doi: https://doi.org/10.31673/2412-4338.2020.022930
106. Kyrychok, R. V. (2018). Test na pronyknennia yak imitatsiinyi pidkhid do analizu zakhyshchenosti korporatyvnykh informatsiinykh system. Suchasnyi zakhyst informatsii, 2 (34), 53–58.
107. Kyrychok, R. V., Skladannyi, P. M., Buryachok, V. L., Hulak, H. M., Kozachok, V. A. (2016). The problems of controlling the security of corporate networks and solutions. Naukovi zapysky Ukrainskoho naukovo-doslidnoho instytutu zviazku, 3, 48–61.
108. Yevseiev, S., Aleksiyev, V., Balakireva, S., Peleshok, Y., Milov, O., Petrov, O. et al. (2019). Development of a methodology for building an information security system in the corporate

research and education system in the context of university autonomy. Eastern-European Journal of Enterprise Technologies, 3 (9 (99)), 49–63. doi: https://doi.org/10.15587/1729-4061.2019.169527

109. Barabash, O., Dakhno, N., Shevchenko, H., Sobchuk, V. (2019). Unmanned Aerial Vehicles Flight Trajectory Optimisation on the Basis of Variational Enequality Algorithm and Projection Method. Proceeding. Actual Problems of Unmanned Aerial Vehicles Developments. Kyiv, 136–139. doi: https://doi.org/10.1109/apuavd47061.2019.8943869
110. Asrorov, F., Sobchuk, V., Kurylko, O. (2019). Finding of bounded solutions to linear impulsive systems. Eastern-European Journal of Enterprise Technologies, 6 (4 (102)), 14–20. doi: https://doi.org/10.15587/1729-4061.2019.178635
111. Barabash, O. V., Musienko, A. P., Sobchuk, V. V., Lukova-Chuiko, N. V., Svynchuk, O. V.; Sadovnichiy, V. A., Zgurovsky, M. Z. (Eds.) (2020). Distribution of Values of Cantor Type Fractal Functions with Specified Restrictions. Contemporary Approaches and Methods in Fundamental Mathematics and Mechanics. Cham: Springer, 433–455. doi: https://doi.org/10.1007/978-3-030-50302-4_21
112. Samoilenko, A. M., Samoilenko, V. G., Sobchuk, V. V. (1999). On periodic solutions of the equation of a nonlinear oscillator with pulse influence. Ukrainian Mathematical Journal, 51 (6), 926–933. doi: https://doi.org/10.1007/bf02591979
113. Svynchuk, O., Barabash, A., Laptiev, S., Laptieva, T. (2021). Modification of query processing methods in distributed databases using fractal trees. Information Security And Information Technologies. Kharkiv – Odesa, 32–37.
114. Laptiev, O., Lukova-Chuiko, N., Laptiev, S., Laptieva, T., Savchenko, V., Yevseiev, S. (2021). Development of a method for detecting deviations in the nature of traffic from the elements of the communication network. Information Security And Information Technologies. Kharkiv – Odesa, 1–9.
115. Lukova-Chuiko, N., Herasymenko, O., Toliupa, S., Laptiev, S., Laptieva, T., Laptiev, O. (2021). The method detection of radio signals by estimating the parameters signals of eversible Gaussian propagation. 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT), 67–70. doi: https://doi.org/10.1109/atit54053.2021.9678856
116. Savchenko, V., Akhramovych, V., Dzyuba, T., Laptiev, S., Lukova-Chuiko, N., Laptieva, T. (2021). Methodology for Calculating Information Protection from Parameters of its Distribution in Social Networks. 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT), 99–105. doi: https://doi.org/10.1109/atit54053.2021.9678599
117. Sobchuk, A., Haidur, H., Laptiev, S., Laptieva, T., Asrorov, F., Perehuda, O. (2022). Modified Fourier transform for improving spectral analysis of radio signals. Modern information, measurement and control systems: problems, applications and perspectives'2022. Antalya.
118. Laptiev, O., Tkachev, V., Maystrov, O., Krasikov, O., Open'ko, P., Khoroshko, V., Parkhuts, L. (2022). The method of spectral analysis of the determination of random digital signals. International Journal of Communication Networks and Information Security (IJCNIS), 13 (2), 271–277. doi: https://doi.org/10.17762/ijcnis.v13i2.5008

119. Tanasiuk, Yu. V., Melnychuk, Kh. V., Ostapov, S. E. (2017). Development and research of cryptographic hash functions on the basis of cellular automat. Systemy Obrobky Informatsii, 4 (150), 122–127. doi: https://doi.org/10.30748/soi.2017.150.25
120. Tanasyuk, Y., Perepelitsyn, A., Ostapov, S. (2018). Parameterized FPGA-bazed implementation of cryptographic hash functions using cellular automata. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 225–228. doi: https://doi.org/10.1109/dessert.2018.8409133
121. Applebaum, B., Ishai, Y., Kushilevitz, E. (2010). Cryptography by Cellular Automata or How Fast Can Complexity Emerge in Nature? Proceedings of the 1st Symposium on Innovations in Computer Science (ICS 10). Beijing. Available at: http://www.wisdom.weizmann.ac.il/~bennyap/pubs/CA.pdf
122. Ostapov, S., Val, O., Yanushevsky, S., Chyzhevsky, D.; Rostanski, M., Pikiewicz, P., Buchwald, P. (Eds.) (2015). Cryptography on the Base of Cellular Automata. Internet in the Information Society. Scientific Publishing University of Dabrowa Gornicza, 71–86.
123. Wolfram, S. (2002). A New Kind of Science. Wolfram Media, 1197.
124. Zhikharevich, V. V., Ostapov, S. E. (2009). Modelirovanie protcessov samoorganizatcii i evoliutcii sistem metodom nepreryvnykh asinkhronnykh kletochnykh avtomatov. Komp'iuting, 8 (3), 61–69.
125. Zhikharevich, V. V., Shumylyak, L. M. (2013). Use of continuous cellular automata for simulation of thermal conductivity in systems with first order phase transition. International Journal of Computing, 12 (2), 142–150. doi: https://doi.org/10.47839/ijc.12.2.595
126. Shumyliak, L. M., Zhykharevych, V. V., Ostapov, S. E. (2018). Application of the asynchronal cellular automata method in the heat conductivity problems investigation. Systemy Obrobky Informatsii, 1 (152), 74–79. doi: https://doi.org/10.30748/soi.2018.152.11
127. Hazdiuk, K., Zhikharevich, V., Ostapov, S. (2020). Simulating Self-Regeneration and Self-Replication Processes Using Movable Cellular Automata with a Mutual Equilibrium Neighborhood. Complex Systems, 29 (4), 741–757. doi: https://doi.org/10.25088/complexsystems.29.4.741
128. Zhykharevych, V. V., Matsiuk, N. O. (2016). Solution the routing problem by modified ant-cellular automaton algorithm. Visnyk ekonomichnoi nauky Ukrainy, 1, 49–54.
129. Clarridge, A., Salomaa, K. (2009). A Cryptosystem Based on the Composition of Reversible Cellular Automata. Language and Automata Theory and Applications, 314–325. doi: https://doi.org/10.1007/978-3-642-00982-2_27
130. Val, O. D., Zhikharevich, V. V., Ovchar, R. I., Ostapov, S. E. (2015). Development and Investigation of the Key Stream Generators on the Base of Cellular Automata. Radio Electronics, Computer Science, Control, 3 (34), 58–63. doi: https://doi.org/10.15588/1607-3274-2015-3-7
131. Rukhin, A., Sota, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S. et al. (2010). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applica-

tions. NIST Computer Resource Center, SP 800-22 Rev.1a, Available at: https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final

132. Dworkin, M. J. (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. doi: https://doi.org/10.6028/nist.fips.202

Edited by
Serhii Yevseiev, Yuliia Khokhlachova, Serhii Ostapov, Oleksandr Laptiev

MODELS OF SOCIO-CYBER-PHYSICAL SYSTEMS SECURITY

Serhii Yevseiev, Yuliia Khokhlachova, Serhii Ostapov, Oleksandr Laptiev, Olha Korol, Stanislav Milevskyi, Oleksandr Milov, Serhii Pohasii, Yevgen Melenti, Hrebeniuk Vitalii, Alla Havrylova, Serhii Herasymov, Roman Korolev, Oleg Barabash, Valentyn Sobchuk, Roman Kyrychok, German Shuklin, Volodymyr, Akhramovych, Vitalii Savchenko, Sergii Golovashych, Oleksandr Lezik, Ivan Opirskyy, Oleksandr Voitko, Kseniia Yerhidzei, Serhii Mykus, Yurii Pribyliev, Oleksandr Prokopenko, Andrii Vlasov, Nataliia Dzheniuk, Maksym Tolkachov

Monograph

Technical editor I. Prudius
Desktop publishing T. Serhiienko

Signed in print 29.05.2023. Format 60×84/16. Offset paper
Digital printing. Typeface EuropeCond. Conventional printing sheets 10.5
Circulation of 300 copies. Order No. 4м-04-8-2023. Negotiated price

---

PC TECHNOLOGY CENTER
Published in May 2023
Enlisting the subject of publishing No. 4452 – 10.12.2012
Address: Shatylova dacha str., 4, Kharkiv, Ukraine, 61165
